CAPE

Construction Application Protocol for Data Transfer: A Building Data Model

Che Wan Fadhil Che Wan Putra

T.I.M.E. Research Institute
Department of Surveying
University of Salford, Salford, UK

Submitted in Partial Fulfilment of the Requirement of the Degree of Doctor of Philosophy, August 1998

Table of Contents

Chapter 4: Data Exchange and Standards

Chapter 5: Product Modelling and the Integrated Environment

List of Tables

List of Figures

Acknowledgements

First of all, I would like to thank all those who helped with the work undertaken in this study especially to:-

- The Government of Malaysia and my employer, Universiti Teknologi Malaysia for providing me the financial support and the opportunity to further my study.

- My superior supervisor, Prof. Dr. Mustafa Alshawi, for his valuable guidance, unfailing support, encouragement, readiness to help and many things, in which without them this study can never succeed.

- Thanks to all members of the AIC Group for their helpful instructions, discussion, and comments in the development of this study especially to Dr. Ihsan Faraj, Dr. Jason Underwood, Dr. Mohamad Jamil Sulaiman, Dr. Zainudin Hassan and last but not least to Dr. Ghassan Aouad for his comments during weekly workshop.

- Dr. Syed Alwee Syed Alsagof for his comments while preparing this thesis and also to Dr. Mushairy and his wife for their proof reading of the thesis.

Finally, I would like to dedicate my thanks to my lovely parents especially my wife, Rosmina, for her patience and continuous support over the years, my sons, Fareez and Farhan and my daughter, Farhaana, for their love.

Abbreviations

AEC	Architectural Engineering and Construction
AIC	Automation and Integrated Construction
ANSI	American National Standard
APs	Application Protocols
BCCM	Building Construction Core Model
BQ	Bill of Quantity
B-Rep	Boundary Representation
CAD	Computer Aided Design
CAPE	Construction Application Protocols for Comprehensive Data Transfer
CE	Concurrent Engineering
CIC	Computer Integrated Construction
CIM	Computer Integrated Manufacturing
COMBINE	Computer Models for the Building Industry in Europe
CONPLAN	Intelligent Construction Planning Generator for Design Rationalisation
CONVERT	Construction Virtual Environment
CSG	Constructive Solid Geometry
DBMS	Database Management Systems
DDE	Dynamic Data Exchange
DETR	Department of Environment, Transport and the Regions
DFD	Data Flow Diagram
DXF	Data Exchange File
EDI	Electronic Data Exchange
ERD	Entity Relationship Diagram
EVALUATOR	Project Estimate and Interim Valuations (Monthly) Generation in an Integrated Environment
IAI	International Alliance for Interoperability
ICAM	Integrated Computer Aided Manufacturing

ICE	Integrated Construction Environment
ICON	Integration of Construction Information
IDEFØ	ICAM Definition Method Zero
IFCs	Industry Foundation Classes
IGES	Initial Graphics Exchange Specifications
INTESITE	Intelligent Site Layout Planning
ISO	International Standard Organisation
IT	Information Technology
NBA	The National Building Agency
OFD	Object Flow Diagram
PM	Product Model
PMI	Project Management Institute
RIBA	Royal Institute of British Architects
SADT	Structured Analysis and Design Technique
SPACE	Simultaneous Prototype for an Integrated Construction Environment
STEP	Standard for Transfer and Exchange of Product Data Model
VR	Virtual Reality

Abstract

Construction is a process, which involves diverse parties having different professional skills and interest. At present, the co-operation and information exchange between parties involved in any construction has not yet been attained. During a project life cycle, the amount of information generated and exchanged is enormous even for a small-size construction project. Current process of managing information flow in construction still lags behind other industries such as manufacturing.

In the era of information age, information technology (IT) becomes a vital tool for managing information. It allows a user/manager to store and retrieve information easily, quickly, produce complete and accurate response, and be better informed of the relevant issues. However, the progress of IT in the construction industry relies on the ability of the project participants to exchange and share information among themselves. Inevitably, there is a need for common standards and approaches due to the lack of compatibility of the information exchange.

The complexity and vast amount of information involved in any construction project and the lack of standards have made the process of producing an integrated environment very difficult. A framework for establishing an Integrated Construction Environment (ICE) has been proposed with the aim of co-ordinating the integration process between the various construction applications. SPACE (Simultaneous Prototyping in An integrated Construction Environment) has been developed which aims to integrate design and construction throughout the project's life cycle via a single database.

The implementation of this framework has led to the development of a modularised central core whereby each application has its own data module. CAPE (Construction Application Protocol for data transfEr) is a design application, which has been developed as part of data module in SPACE. It represents a building elements data module and the object interpreter engine. It aims is to improve the flow of information between project's participants, particularly those related to the design stage.

The development of CAPE data module has resulted in the implementation of a system, which capture most of the design elements in CAD (AutoCAD/AEC^{TM}), the study of their properties such as co-ordination and dimension and populates it into the object-oriented database systems to serve other application modules in the project model, i.e. SPACE. CAPE data module also provides several benefits. It provides essential support for the integration of design and construction, a generic set of building element classes, defines building elements with the necessary information at run-time, and a dynamic and an independent environment for all graphical packages such as CAD and VR.

Chapter 1

Introduction

1.1 Introduction

Construction projects proceed through a series of phases during their life [Sanvido, 1992a]. These project life cycle phases have been classified as manage, plan, design, construct, and operate functions [Sanvido *et al.*, 1990]. Information is one of the key elements that drive these processes. During a project life cycle, the information is passed from one phase to another. In a large project, there are substantial volumes of information flow between the various players during the design and construction of a particular project [Watson, 1995]. As a result of this, there is an increase of information processing. Sanvido [1992a] has highlighted some of the major factors, which led to the increase of information processing requirements for the project team as follow;

- The market place often dictates that projects are required in increasingly shorter time periods. This typically requires concurrent design and higher staffing levels. These conditions increase information flow among the project team.

- The product requirements may be changed several times during a project's life cycle in order to serve the client's need. As a result, there is an increase in the information processing requirements whereby all these changes need to be communicated to the specialists at various stages of the process.

The information processing in the construction industry, which involves diverse parties having different professional skills and interest [Munns, *et al.*, 1995], requires the parties to co-operate and exchange information in order to complete the building project. However, this is not yet successfully implemented. Many participants are involved in the construction project such as contractors, sub-contractors, suppliers, etc. rarely undertake their responsibility on site [Keijer, 1992]. These participants tend to operate from their perspectives and not from the whole project perspective, thus resulting in local optimal solutions. Furthermore, many participants who specialised in construction and have different views of the process have drifted apart and are often geographically isolated from each other [Aouad, *et al.*, 1994]. These have resulted in the fragmentation of the construction industry itself [Howard *et al.*, 1989]. The fragmentation exists both within individual phases of the construction process (e.g. the design phase), as well as across project phases from planning through design and construction and into facility maintenance and operation. Brandon and Betts [1995] stressed that:-

> *"fragmentation refers to the fact that the different participants in the design and construction of buildings have been trained, and gained experience, through different educational and professional paths, are employed in distinct organisations, have distinct value systems and methods of working, and have derived separate information management systems. Information management, because of this, is highly fragmented"*

As a result of the fragmentation of the AEC industry, generating, sharing, and maintaining project information throughout a project life cycle constitute a complex assignment [Howard *et al.*, 1989]. Several professionals from various organisations receive and utilise project information, producing additional information and contributing to the evolution of AEC projects [Kartam, 1994]. These numerous numbers of professionals involved in several stages of an AEC project have led to decreased integration, co-ordination and collaboration among the participants. As a consequence of the lack of industry integration, professionals responsible for one specific AEC project phase depend on information relative to other phase to be able to perform their functions. These lead to problems, for instance, incomplete specifications and design, changes in site conditions, misinterpretation of contract documents, unconstructable designs, etc.

Both horizontal and vertical flows of information are essential for lessening prejudicial effects of industry fragmentation [Kartam, 1994]. Horizontal flow of information is the communication among professionals of different disciplines involved in one specific project phase. For instance, during the design phase, structural and electrical engineers, along with several other professionals, are working simultaneously on different plans of the same project and the communication between them are indispensable. Vertical flow of information, on the other hand, means the exchange of information between professionals of different disciplines involved in different project phases. For instance, communication between construction managers and architects is fundamental for project accomplishment. Therefore, extensive flow of information, i.e. both horizontal and vertical is vital aspect when trying to increase industry efficiency.

Construction is an information intensive industry that will gain large benefits from the automation of its information flow and the creation of shared knowledge [DETR, 1996a]. By sharing the information, the construction industry will gain many benefits such as increasing productivity, improving quality, etc. [Tah *et al*, 1994]. However, current approaches of information sharing in the construction industry suffer many problems. These problems arise due to the bulk of these information flows which are still in the form of traditional paper drawings [Watson, 1995]. This often leads to the difficult flow of information, substantial loss in efficiency of design, planning, and construction phases of a project, as well as increased potential for errors [Kartam, 1994]. A recent study by DETR [1996a] also highlighted some of these problems such as:-

- Difficulty in codifying information;
- Loss of information at the end of projects;
- Lack of facilities to store information;

Today, in the emergence of information technology (IT), most project information is stored and processed on computers. These may include CAD systems, structural analysis/design packages, project management packages, etc. However, the presence of these software packages tend to solve specific problems and business needs only. Most of the packages are stand-alone which results in 'island of automation' within the industry. Therefore, by integrating project participant's computer systems, the effectiveness of a project team will be greatly enhanced, i.e. by increasing data sharing, reducing time requirements and errors for data input and output, accelerating communication among participants, and improving the completeness of information received by each team member [Fischer & Froese, 1992].

The word 'integration' has become very widely used to describe the desirable concept of freely exchanging information between different participants in the construction process, yet actual examples of integration are relatively limited and localised [Vincent, 1995]. Today, effective integration requires the continuous and interdisciplinary sharing of project goals, data, and knowledge among all project participants [Fischer & Froese, 1992]. Such integration requires an agreed framework whereby the information can be integrated. Many researchers believe that such framework can be achieved through an integrated environment [Yamazaki, 1992; Faraj & Alshawi, 1996; Björk, 1992; Teicholz & Fischer, 1993; Iosifidis *et al.*, 1995] whereby all possible construction applications can be integrated under one environment. However, the complexity and the vast amounts of information involved in any construction project and the lack of standards, have made the processes of producing an integrated environment very difficult [Sanvido, 1992b; Faraj & Alshawi, 1996; Aouad & Price, 1993]. Until now, the question of how such framework of an integrated environment should be structured and implemented effectively still remains to be further investigated.

1.2 The research background

The construction industry by nature is a transient and highly fragmented industry and because of these two characteristics, co-ordination and proper communication have significant effect on productivity and quality of construction project [Sadri & Kangari, 1993]. In the construction industry, the planning, design and construction activities are typically carried out by participants in different organisations. These various stages are described in R.I.B.A Plan of Work [RIBA, 1980], whereby each stage of the project life cycle contain processes which are executed by a specific profession, i.e. designers, contractors, etc. Due to the large number participants involved in the project life cycle, the lack of proper communication, information sharing and exchange exist. During the last three decades, there have been several attempts to find the solution for the above mentioned problems. A large number of researchers looked for ways of improving the communication between the different parties involved and in particular, to enhance the flow of a project's information. In recent years, integrated systems and information integration have shown to have many promises towards solving complex problems such as those in the construction industry [Howard *et al.*, 1989; Aouad *et al.*, 1994; Fischer & Froese, 1992].

'Integration' is sometimes quoted as having happened when any substantial amount of information has been exchanged in digital form [Vincent, 1995]. Exchanging electronic drawings, for example, may actually offer little improvement over exchanging paper drawings whereby it still requires a skilled eye to interpret them. Historically, the UK construction industry has been reluctant to take the initiative regarding moves towards the integration of construction information [Watson, 1995]. Although many different application programs are in use, there has been relatively little provision for transferring information between them. The consequences of this result from two folds, i.e. the contractual nature of the construction industry and the 'islands of automation' in which each application's programs are isolated between each construction discipline.

Several efforts have been devoted towards the development of a framework for an integrated environment in the construction industry. However, the successful development and implementation of Computer Integrated Manufacturing (CIM) (improving time, cost and quality) has led to the development of Computer Integrated Construction (CIC). CIC consists of both of the use of information technology in the different phases and tasks in construction, and the integration of these phases and tasks through the use of digitally stored data and data transfer [Björk, 1992]. CIC can address the needs and requirements of all parties involved in a project life cycle [Alshawi, 1993].

It has been recognised widely that in order to achieve the best performance, such integrated systems need to share and exchange data through a single core, i.e. a product model [Alshawi & Che Wan Putra, 1995]. The product model is the core of any CIC environment whereby a library of objects is formed from which a symbolic project model can be built up [Levitt *et al*, 1991]. Such a symbolic model can capture the various components of a project such as walls, beams, etc., their properties e.g. thickness, length, etc., and their relationship with each other e.g. embedded in, supported by, attached to, etc.

Although the industry have witnessed the integrated systems to be the better solution of the fragmented construction, the main problem would be the compatibility of the integrated information generated between each participants. Therefore, an agreed standards are required for the integration of the various disciplines involved. The most important is the data exchange format. Although the issue of a standard neutral data exchange format has started since late 70's, the initiative of establishing the international standards for exchanging data between applications such as STEP by the International Standard Organisation (ISO) at late 80's has been welcomed world-wide, especially in the construction industry. However, these standards have been materialised in the form of data modelling and data representation and not into protocols for data transfer [Alshawi & Che Wan Putra, 1995]. Application Protocols (APs) committees have been established to meet the increasing demand for establishing a format to transfer project specific information. However, implementation problems are quite different from theoretical data modelling and can reveal significant weaknesses in any theoretical development. Therefore, best results can only be achieved through a simultaneous consideration of theoretical data modelling and practical implementation.

In order to overcome the above-mentioned problems, two main issues which are related to information management and information flow within and among the various data models in an integrated environment i.e. object definition and data sharing are studied. Object definition is the most efficient number of attributes an object can be populated with, in order to serve all possible construction applications efficiently and effectively. While shared data is a common data, which serves the interest of a number of objects in the integrated environment. These two issues are highly inter-dependent, i.e. the amount of data stored in each object can be significantly effected by the concept of data sharing.

1.3 Research hypothesis

The research hypothesis of this study is:

> " Is it possible to capture the project information, model it and be able to manipulate it in a single integrated database?"

This hypothesis will only be concerned with project information that are necessary to serve major stages of the project life cycle such as design, construction planning, estimating and site layout planning.

1.4 Aims of the research

The primary aim of this research is to improve the flow of information between project's participants, particularly those related to the design stage. Such a flow of information can be facilitated through a central database which can serve as a repository for project information whereby all project participants can read from and write to. This can be implemented through the integration of CAD packages with an object oriented environment where project information can be accessed, in a structured manner, by other applications such as construction planning, Virtual Reality, site layout planning, estimating, etc. CAD drawing can be dynamically interpreted into meaningful objects while building elements data module can be established to support multiple designer views and to serve all downstream applications.

1.5 The objectives of the research

The objectives of this study have been divided into four parts and are summarised as follows:

1. Develop a methodology and a framework for an integrated construction environment through:-

    (a) Gaining an understanding and identifying the problems related to the information management and information flow in the project life cycle.

    (b) Investigating the potential of information technology (IT) for the integrated environment, i.e. the possibilities of integrating the applications under one environment.

    (c) Investigating the problems of implementing data models in integrated environments.

    (d) Proposing a framework related to information sharing and information flow within and among the various data models in an integrated environment considering the object's definition and data sharing within the life cycle.

2. Develop an object interpreter engine (OIE) in AutoCAD-AEC™ through :-

    (a) Developing an algorithm for object interpreter engine which will be used for translating AutoCAD-AEC™ drawing primitives to objects.

    (b) Developing an algorithm for the topological relationship between the objects in

AutoCAD-AEC™.

(c) Developing an algorithm for the graphical file - DXF (AutoCAD Data Exchange Format) for exchanging the object information to other graphical applications such as VR and other CAD packages.

(d) Developing an algorithm for analysing spaces in order to identify the space boundaries and space separator. This will provide the functionality for functions such as heat lost (if required), total area for surface finishes, etc.

(e) Transferring the interpreted objects into the building elements data model in an object oriented environment tool.

3. Develop a building element data model in an object-oriented environment through:-

(a) Developing a conceptual model for the building elements which could cater for the design information and those required by other construction applications.

(b) Mapping the data structure of the developed conceptual model into an object oriented environment tool.

(c) Developing an algorithm for the knowledge in the data model to cater for:-

- Topological relationships e.g. attached to, embedded in, supported by, etc.
- Type of building elements e.g. simply supported or continuous beams.

(d) Incorporating the prototype into the single integrated environment (SPACE) whereby it can be shared by other construction applications such as construction planning, site-layout, estimating, etc.

4. To evaluate the prototype and suggest recommendations for future research.

1.6 Methodology of the research

In order to achieve the above mentioned aims and objectives, an extensive work has been carried out. The general outline of the research methodology framework is shown in Figure 1.1 where the research methodology has been divided into four phases. The details of the work involved are as follows:-

1. Literature review has been conducted considering most of the previous research done in the area under investigation. Information were collected from books, journals, reports, conference proceedings, etc. Through this review, the domain area is defined.

2. Once the domain area has been reviewed, the problem area is identified. In order to achieve the aims and objectives of the study, the scope and the methodology of the research need to be defined.

3. While developing the prototype model, the conceptual models for the domain area, incorporating the multiple views of the integrated construction environment, are developed. The structure framework for this environment is proposed by representing the generic activities along with their relationships using the process model. Later, the information required for the development of an integrated construction environment is modelled in which it represents the building elements data model. These models are then mapped into the project model in the object-oriented environment.

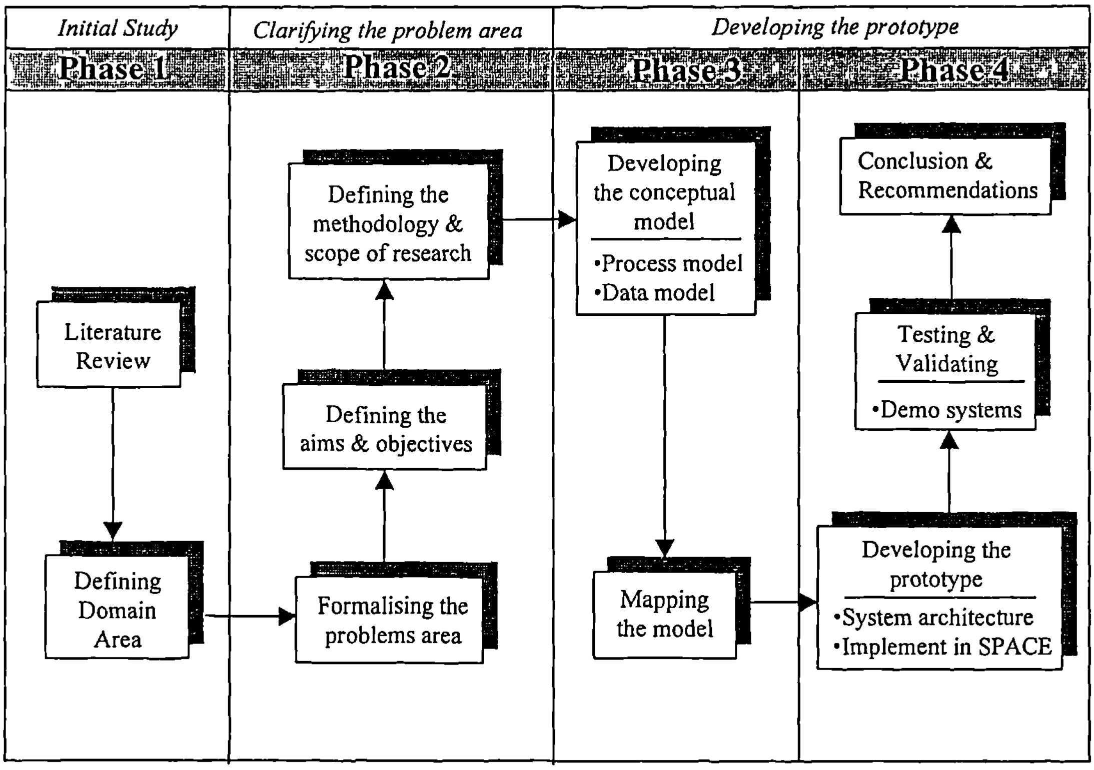


Figure 1.1: Research methodology

4. The prototype is developed and implemented into the SPACE environments. Two steps have been taken, i.e. developing an object interpreter engine in the CAD system and a building elements data model in an object-oriented environment. The testing and validating of the prototype systems are done through a number of demonstrations.

5. Finally, conclusions are drawn and the future developments of the research are recommended.

1.7 Scope of the research

The main aims of this study is to establish a building elements data model which support multiple designer views in order to serve all downstream construction applications. The developed prototype system should incorporate most of the required design information. However, the development and implementation stages of this work are limited to the following:-

1. The design elements are limited to reinforced concrete office building.

2. The number of design elements interpreted is limited to cavity wall, solid wall, window, door, beam, column, slab and pad footing only.

3. The building elements data model contains limited views. For example, a column is viewed in terms of its shape not in terms of its design. The design views have been implemented separately in other data modules.

4. The topological relationships are limited to three types only, i.e. supported by, attached to and embedded in. This is due to the complexity of identifying the topological relationships using "wire frame" based CAD systems.

5. The implementation of the prototype is limited by the software used, i.e. AutoCAD-AEC™ and KAPPA-PC™.

6. The design information which are captured, transferred and manipulated to the central core, served only the applications developed concurrently.

1.7 Guide to the thesis

This thesis has been divided into eleven Chapters. This Chapter has identified the fragmentation issue of the construction industry and stressed the needs of a structured framework of an integrated construction environment whereby all the design and construction applications can be integrated. The remaining Chapters are organised as follows:-

Chapter 2 highlights the information flow involved in the construction process which needs to be managed to ensure smooth communication between the participants in the project life cycle. It also highlights that the application of IT in the construction industry can be use very effective if it has the ability to exchange and share information among the project participants.

Chapter 3 examines the issue of information sharing, the concept and the implementation. The approaches to integration and the current attempts towards the development of an integrated environment in construction and other industries are highlighted.

Chapter 4 examines the issue of data exchange and the standards with special reference to the construction industry. The development of international standards, STEP and the latest development of Industry Foundation Classes (IFCs) by the IAI is investigated.

Chapter 5 discusses the product modelling within the integrated environment in which the information modelling approaches have been explained. A brief description of product models developed in an integrated environment such as RATAS, ICON, ATLAS, GenCOM, COMBINE, OSCON and WHISPER are reviewed.

Chapter 6 proposes a strategic, but generic, framework for establishing an integrated construction environment. The typical product model of the proposed framework at various levels of abstractions whereby a full process analysis of the various activities are required to successfully establish such an environment is explained. The implementation of the ICE, i.e. SPACE has been described including its components, i.e. CAPE, SPECIFICATION, CONPLAN, EVALUATOR, INTESITE and CONVERT.

Chapter 7 discusses and highlights the issue and the development of data sharing in an integrated construction environment using object definition. The problems with the implementation of data models by separating the theoretical and implementation issues are highlighted. The importance of clearly defined objects and their attributes are addressed. The structured concept of object's life cycle, which aims at managing information and its flow within the integrated environments is also proposed.

Chapter 8 describes the data models, which represent the information required for the development of an integrated construction environment including the development of the "building elements data module" using EXPRESS-G modelling technique. The data models have been decomposed at various levels of abstractions. The relationship between the "building elements data module" with other modules is discussed by stressing the impact of the "building elements data module" in the development of other modules in the ICE.

Chapter 9 describes the implementation of the prototype and how the conceptual data model is mapped into the object-oriented environment. The development of the prototype have been divided into three main parts i.e. the graphical interface, the CAD system in the AutoCAD-AEC™ and the knowledge-based object-oriented database in KAPPA-PC™.

Chapter 10 describes the evaluation of the developed prototype. A demonstration presentation has been carried out to determine the system's performance, capabilities and its limitations. The testing procedure and the results from integrating the prototype with other applications are also discussed.

Chapter 11 presents the summary and conclusions of the research work together with the recommendations for future work.

Chapter 2

Information Management and the Project Life Cycle

2.1 Introduction

Construction is one of the most information dependent industry which obtains information from detailed drawings, cost analysis sheets, budget reports, risk analysis charts, contract documents, etc. [Tucker & Mohamed, 1996]. During project life cycle, the amount of information generated and exchanged is enormous even for a small-size construction project. Therefore, information management which can highly influence cost, time and quality becomes an important issue. Many authors believe that information management can have a significant impact on the success and profitability of the entire industry [Atkin, 1990; Vanier *et al*, 1993; Fischer *et al*, 1993].

Information management has been accepted as an essential management discipline in the manufacturing, aerospace and defence industries for many years. These industries spend over 0.5% of their turnover on information management [Atkin, 1990]. However, there is no comparable figure in the construction industry and it is clearly still lags behind. The reason behind this is that the construction industry is made up of a large number of very small organisations. For example in UK, there are approximately 200,000 companies [DETR, 1996a]. This therefore causes problem in tabulating the data.

This chapter presents the project life cycle stages, highlighting the role of professionals involved in the project life cycle. The issues of information management over the project life cycle and the needs for and problems of managing the project information in the wake of the information technology (IT) era are also discussed. Finally, the process of improving the management of project information flow is highlighted.

2.2 Project life cycles stages

Generally, a project (construction or otherwise) is a unique undertaking for essentially a single purpose which is defined by scope, quality, time, and cost objectives [Ahuja *et al*, 1994]. A project occurs over an identified period of time during which a changing level of effort is required to complete each stage. A project has also been defined as the process of working to achieve a goal [University of Minnesota, 1997]. During the process, projects pass through several distinct phases, which are called the *project life cycles*. Meanwhile, in a construction project, a project life cycle can be viewed as a project proceeding sequentially through five distinct stages starting from the conceptual to the occupation/maintenance [Alshawi, 1996]. It includes conceptual, design and tender as the pre-contract stages whereby the construction and occupation/maintenance as the post-contract stages as shown in Figure 2.1.

In the traditional method of procurement, the pre-contract stage is the most important stage in a project life cycle, whereby the contract is signed and tendered. It is at this stage where decisions are made that can significantly affect the immediate and long-term construction problems. The post-contract stage is the stage where a contractor is appointed and the construction process is carried out. This is the stage where all the documentation (bill of quantities and general arrangement drawings with detailed drawings and specifications) are supplied to the contractor.

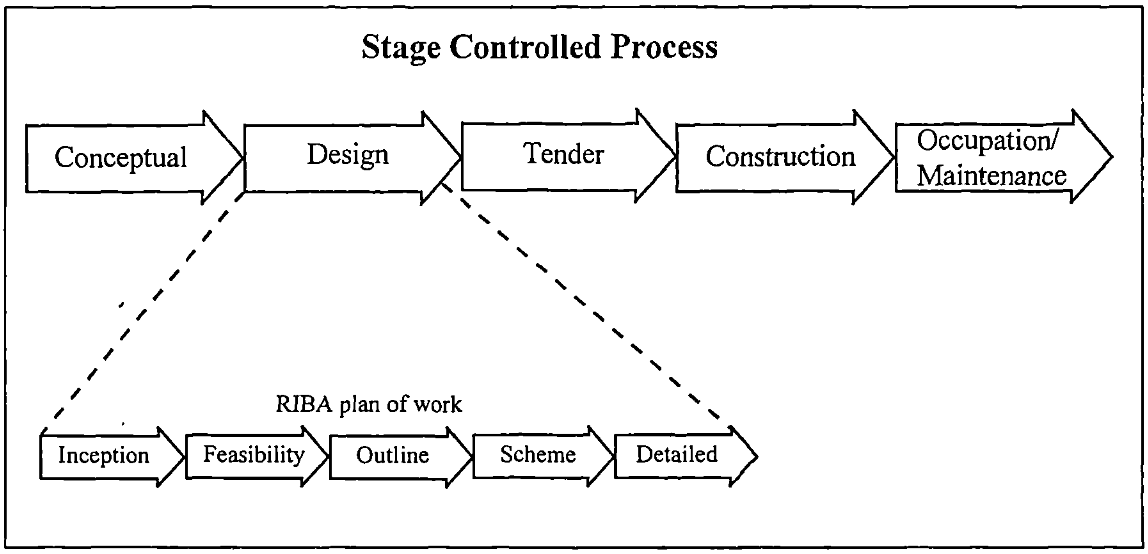


Figure 2.1: Traditional project life cycle

The following sections briefly describe the five stages of the project life cycle within the traditional procurement. It is outside of this study to explore such stages and to examine the flow of the information between them.

2.2.1 Conceptual stage

The purpose of this stage is to establish the client's outline requirements, that is, to decide what is required [Ward, 1979]. Such requirements are time/duration of the project, cost of the project, construction reliability and performance and management. The conceptual stage seeks to determine whether the project is capable of execution in terms of its physical complexities, planning requirements and economics [Harvey & Ashworth, 1993]. At this stage, the client must carefully investigate all alternative sources before proceeding to the next stage.

2.2.2 Design stage

The design stage is a cyclical process with a linear progression, which in itself comprises of several distinct phases [Alshawi, 1996]. Due to its nature, a guideline has been proposed by the RIBA (Royal Institute of British Architects). This "RIBA plan of work" (Figure 2.1) is a model procedure which provides a process for the design of any project. It is not rigid and can be modified to meet the individual circumstances that prevails on a particular project [Ward, 1979].

Initially, the design commences with the design brief, with the aim of establishing the client's outline requirements. The client must make the decision to build, appoint an architect, and consider the outline proposals for the project brief. The next stage is to do a feasibility study, which aims at studying all technical, functional and financial aspects associated with the project, and advising the client as to whether his proposal is feasible. If the client agrees to proceed with the project, the next stage will be the outline proposals. The purpose of this stage is to examine alternative proposals in order to determine the general approach to the layout, design and construction of the project [Ward, 1979]. This process involves collaboration between the designers and the client.

During the scheme design stage, the wider issues of appearance, method of construction, outline specification and financial matters are examined in detail. At the end of this stage, the architect reviewing the evolving project design to date will produce a report. This report is presented to the client for approval. Once the client has approved the scheme, the design brief cannot be altered. The design of the scheme is developed in depth in the detailed design stage. Final decisions are taken to facilitate the full design of all parts and components of the building. At this stage, the designers add low level of detail to the project's conceptual design, i.e. the project takes its final form, such as detailed structural design, foundation design, services design, etc.

The production information is the final stage of the design aiming to expand the design information in order to give sufficient details for the project to proceed. The production information includes production drawings, specifications, schedules, contractual particulars and bill of quantities.

2.2.3 Tender stage

The purpose of tendering is to assign a contractor to carry out the construction of the project [Ward, 1979]. During this stage, the construction documents are sent to the appropriate contractor(s), depending upon the method of tender. Upon receipt of the documents, the contractor estimates the costs to carry out the construction work and adds an additional percentage for overheads and profit to arrive at a total tender figure [Forster, 1986]. The contractor submits this figure as a bid for the work in competition with other contractors. Traditionally, the contractor who submits the lowest bid is awarded the contract. In addition, the bidding contractors must also submit a proposed construction program (pre-tender program) to show the planned duration of the work [Forster, 1986].

2.2.4 Construction stage

On being awarded a contract, the successful contractor, prior to taking possession of the site, has to enlarge the pre-tender program and produce a long-term construction plan. During site operations the contractor should carry out valuations of each completed stage of work, whereupon interim certificates are prepared so that stage payments may be made to the contractor and sub-contractor to pay for work done to date. It is normal practice for a percentage to be retained to pay for faulty work, which may develop or be subsequently discovered later.

2.2.5 Occupation/Maintenance stage

Once the project has been completed and the work has been inspected and approved by the designer (architect), the project is then handed-over to the client for occupation. With the exception of a defects liability period (usually six months after the hand-over date), where the contractor is responsible for rectifying any defected work, the general maintenance and up-keep of the project now becomes the responsibility of the client.

2.3 Procurement methods and the role of professionals

The professionals are individuals or organisations who are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or successful project completion [PMI, 1996]. They may be clients, project managers, designers (architects, structural engineers, building services engineers), quantity surveyors, contractors, etc. The roles and responsibilities of professionals may overlap because they often have very different objectives that may come into conflict. For example, in the same project, an architect may be keen to look at only aesthetic value, the structural engineers at the safety, the estimator at cost, etc.

The roles of professionals over the project life cycles rely on the co-ordination of all parties involved in a project. This varies depending on the method of procurement which normally implies a certain set of relationships between members of the design team and other professionals [Brandon *et al*, 1988]. The procurement process is undoubtedly the foundation on which perception of the construction industry and therefore, the role of integration is based [Jennings & Kenley, 1996]. There are three main methods of procurement currently being used in the construction industry, i.e. traditional, design and build, and management. In the traditional method of procurement, the client appoints independent consultants to act on his or her behalf to produce the design and supervise the construction. Project delivery is viewed as a sequential process in that the design is largely completed before the appointment of the building contractor to whom detailed plans, specifications and possibly bill of quantities must be given. In the design and build method of procurement, the client makes an agreement with one single administrative entity (the prime contractor) who is given responsibility for the whole project, from initial briefing and design through to construction of the building. While in the management method of procurement, the client appoints a management contractor to work alongside the other professional consultants (thus combining the management and construction role). The aim of this method is to incorporate construction expertise into the design at an early stage in the project [Brandon *et al*, 1988].

2.4 Integration problems

The traditional system of design and construction has led to a number of problems for the industry [Ratcliffe, 1985]: (a) incomplete initial design caused changes in the design during the process of construction and disruption to the building programme; (b) the lack of clear detailed brief specifying the client's requirements at the outset creates uncertainty and misunderstanding among the members of the professional team; and (c) failures to appoint consultants and contractors at early enough stage in the process and distinguish clearly their roles and responsibilities, i.e. choosing the right method of procurement systems, has led to ambiguity and lack of coherence.

This has led to a decay of integration between the various professions and to a misunderstanding of the role of each profession. Several studies have outlined the limitations of the traditional approach to the design and construction process:-

- The design process usually takes a considerable time depending on the size, complexity and nature of the project [Alshawi, 1996];
- Design solutions do not usually meet budgetary constraints especially at early design stages [Ferry & Brandon, 1991];
- Weak communication between members of the design team due to their different design perspectives [Aouad *et al*, 1994];
- Large percentage of construction problems on site are caused by complex designs [Alshawi & Underwood, 1996];

At the design stage, each designer often works independently in a separate location. Thus, the project team is only loosely integrated and this harms the efficiency of the project. These problems are exacerbated by poor communications between the project members and between the parent company and the project site [DETR, 1996b]. These poor communications affect the accuracy and efficiency of many of the stages of the project life cycle, especially those concerned with the early life of the project. At those stages, as explained earlier, better communications and the use of information can produce large saving and could significantly impact the time and quality of the project.

2.5 Project co-ordination, the needs and benefits

In the traditional method of procurement, there are many parties involved directly in a wide range of activities. One of these activities is the translation of the client brief into design or from design into a constructed building. For example, during the pre-tender stage, the architect, engineers, client and cost advisor share and exchange information and data that evolves as the design progresses. At the early stage of project life cycle, such co-ordinations are essential to ensure the success of the project.

Co-ordination is basically a management tool and can be defined as the bringing into a proper relation of the various activities related to a project, or causing these activities to function together, or in a proper order [Jegaraj, 1983]. While, the project co-ordination is the planning of a project, or whole series of projects, well in advance of the start of pre-contract work and the control of the project through all its stages [NBA, 1972]. The main target of the project co-ordination is to provide a smooth flow of information among the project's life cycle stages, i.e. by providing the means for efficient communication [Vienna University of Technology, 1997]. Project co-ordination also adds a new dimension to the project delivery strategy, i.e. the strategy whereby each party involved must be co-ordinated towards a common goal, that is, the timely delivery of an efficient project [Jegaraj, 1983].

Project co-ordination generally leads to an increase in efficiency, which can in turn generates several benefits. A study by NBA (The National Building Agency) in the UK [1972] revealed that such benefits are;

- *Better communication,* - improved appreciation of the work of each participants involved;
- *shorter construction time,* - getting specifications, schedules and details to the contractor at the right time;
- *Earlier occupation,* - earlier completion of the project.

Thus the project co-ordination is a necessary tool in the project delivery strategy whereby the effectiveness of which depends on the creativity of the parties involved in the project life cycle.

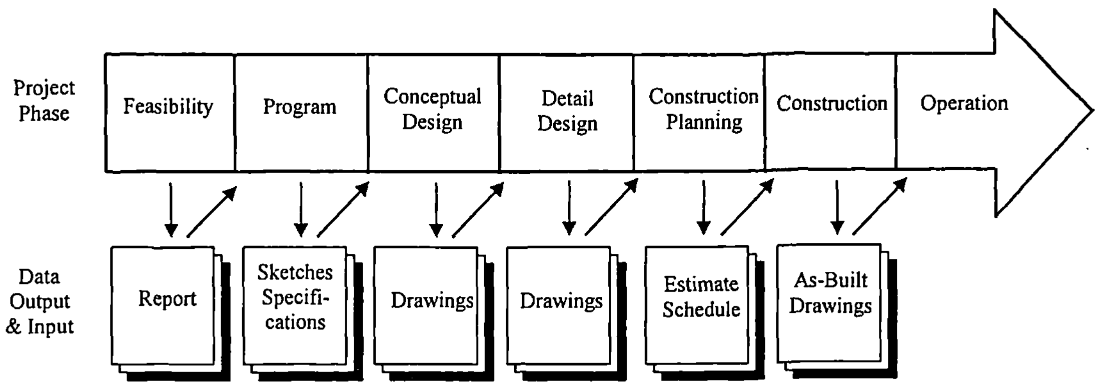


Figure 2.2: Traditional fragmented and sequential project delivery process [Amended from Teicholz & Fischer, 1993]

Figure 2.2 shows the various phases of a project along with their inputs and outputs. It can be clearly seen that the outcome of one phase is the input for the next. This approach, which is usually materialised in the traditional procurement method, enhances the industry fragmentation and makes the co-ordination process between the different professionals difficult to achieve. Data is prone to errors because it is extracted, transferred, interpreted, and repackaged very frequently. Therefore, it is important to manage the flow of information in order to avoid or minimise the problems.

2.6 Project information and information technology (IT)

Project information can be defined as the information that describes the physical facility (product) and is required for managing its process [Australian STEP Demonstration Project, 1995]. A typical set of project information may include site survey, cost analysis, design drawings, specifications, regulations, bill of quantities, project planning, job costing, estimates, valuations, material management, facility management, etc. This information is usually handled by different departments within one organisation or different organisations, resulting in long loops [Alshawi, 1996]. These loops can cause lengthy delays and inconsistencies of data used by different departments.

Large amount of project information are generated and used during the various stages of a project life cycle. Sharing and maintaining this information among multiple disciplines and throughout a project's life cycle is a complex and difficult task [Ito *et al*, 1990]. However, such flow of information needs to be managed so that it will be received or accessed when required. Information management is not for handling drawing issues only, but if it is correctly administered, it can provide the necessary framework whereby everybody knows what his responsibilities are and how to carry them out [Atkin, 1990].

In the era of the information age, information technology (IT) becomes a vital tool for managing the information. IT is a generic term used to describe all types of computer and communication technologies applied in the acquisition, storage and retrieval of information [Betts *et al*, 1989]. In the construction industry, it has been proven that the use of IT could reduce the fragmentation problem [Betts *et al*, 1989; Atkin, 1989]. Currently, however IT can only assist various processes in construction, such as planning, estimating, structural design, etc., and does not automate them.

The large volume and diversity of information used and exchanged among professionals with different backgrounds require the use of IT tools [Betts *et al*, 1989]. Current IT tools provide excellent means of transmitting the syntax of information, but do not as yet have the capability to communicate the "meaning" of the information which are supposed to be conveyed to other participants in the construction process. The construction industry can benefit from the IT tools today through the facilities of storage and manipulation of vast amount of data and their speedy transfer. Various participants in the construction process use IT tools for specific functions in their individual domains, but the information generated by the IT tools is not transmitted in a directly useable form to other participants. However, many studies have suggested that IT has the potential to be the most powerful tool for re-engineering the construction process by improving the performance of traditional processes [Atkin, 1990; Tucker & Mohamed, 1996; Tucker, 1996].

Despite the potential use of IT as a tool for managing the information flow, the supply of information to the industry is still in poor condition. A recent study by DETR (Department of Environment and Transport) UK [1996a] revealed that this problem occurs due to the following;

- Information does not follow in a common format or description and is hard to find and use;
- The source of information to the construction industry is fragmented and diverse;
- Information is often of unreliable quality and is not easily searchable or useable;
- Different access methods and user interfaces make it difficult to use a variety of information sources;

The above statements is also supported by other construction organisations, which also revealed that the underlying causes of the problems in the adoption and utilisation of integrated IT systems, can be attributed to [Alshawi, 1996];

- poor management and communication;
- the fragmented nature of the industry;
- lack of standardisation and uniform procedures;
- the number of participants involved in construction projects;

The problems that the construction industry is facing today are obvious. The process of managing the information flow in the construction process needs to be improved in order to increase the productivity especially in terms of time and cost.

2.7 Managing the flow of information

As discussed earlier, the problem of managing the information is to bring the participants to work and share the information together. The construction industry, which is an information intensive industry, will gain large benefits from the automation of its information flows and the creation of shared knowledge [DETR, 1996a]. By sharing information, the industry will increase its rate of learning and produce a sustainable and increasing profit level. Industry shared information will also create a more efficient market that will improve both industry performance and the quality of the finished product.

The introduction of project central database is one of the solutions for sharing the information. The principle of a database is that it is a single centralised source of information. Such central databases are usually developed as information networks and can be used at either the project, enterprise or industry level. Many different users of information can share access to the same information from the database.

According to Aouad *et al* [1994], the integrated databases are useful for the establishment of a powerful environment for information sharing and integration. The communication between the various parties involved in a construction project will gain the benefits from such databases whereby common data stored in the databases can be shared and the need for re-entering data can be eliminated. In addition, an integrated database environment ensures that each party is working on the right piece of information, thus changes can be recorded, traced and updated.

2.8 Summary

Construction is one of the most information dependent of all industries. Information can be in the form of detailed drawings, cost analysis sheets, budget reports, risk analysis charts, contract documents, etc. The traditional practice in the construction industry has led to a decay of integration between the various professions involved in the project together with a misunderstanding of the role of each profession. During the project life cycle, the amount of information generated and exchanged is enormous even for a small-size construction project. The information flow involved in this process needs to be managed to ensure that it is communicated to whomever needs it, whenever they need it, in whatever form they need it, in order that they meet their objectives.

In the era of information age, information technology (IT) becomes a vital tool for managing the information. Electronic management of information through the use of IT is the most effective way to manipulate information. It allows users/managers to store and retrieve information easily, gain faster, complete and accurate response, and be better informed of the relevant issues. However, the effective use of IT in the construction industry relies on the ability to exchange and share information among the project participants.

The following Chapter will discuss the issues of the information sharing and the steps towards the integrated environment.

Chapter 3

Information Sharing and Integrated Environments

3.1 Introduction

The previous chapter has discussed the issues and the needs of managing information flow within the project life cycles. It highlighted that information sharing and integrated environments are important issues, which can bring about an effective way to manipulate information. This chapter examines the issue of information sharing; the concept and the implementation. The approaches to integration such as the direct translator, standard exchange format, blackboard architecture, and product model are explained. Attempts which are carried out in the construction and the manufacturing industries towards integrated environment such as Computer Integrated Construction (CIC) and Computer Integrated Manufacturing (CIM) are also discussed. Finally, Concurrent Engineering (CE) is presented.

3.2 Information sharing

Information sharing means having each participant in the construction process working from the same shared project model [Howell, 1996]. This shared project model will incorporate the information created by each discipline, then capture the relationships among this information to enable it to be shared by others. Although shared information can be critical to the success of construction projects, organisational and technical barriers can reduce the quality, and accuracy of the shared information between participants [DETR, 1996b].

According to Howell [1996], this sharing of information can bring many benefits to the construction industry. These benefits include:

- *Efficiency* - Access to the industry information defined by previous applications can eliminate much manual data entry;
- *Improved Accuracy* - Current data-translation mechanisms exchange data at the "lowest common dominator" of the applications they serve. These applications interpret this information with a variety of translators or else require manual take-off. This translation frequently causes errors and routinely results in significant loss of information. A shared project model provides a uniform structure for project information, therefore eliminating errors caused by translation and interpretation;
- *Better communication* - A shared project model allows communication of design between members of the building team;

- *Automation* - With a shared project model, application developers are able to automate many processes that are currently manual;

Despite the importance of sharing information, it is found that the implementation of information sharing having many obstacles. The current contractual nature of the construction industry has led to additional implications for the sharing of information [Alshawi, 1996]. Complex information flows, which are inherent in the industry, cannot easily be reduced by process improvement or by changes to contractual arrangement [BT, 1995]. Along the project life cycle, information has to be shared amongst the various contractual parties involved, to enable project to progress successfully. However, the contractual parties believe that it is not in their interest to share the information [BT, 1995]. The information is regarded as a source of power, and is not readily accessed. For example, from the design point of view, the design created by each specialist design team becomes the "rightful possession" of the designer, because of the time and expertise exerted in developing it. This may cause a certain amount of reluctance from a designer having to share/exchange some of this information with other parties involved.

Table 3.1 summaries the type of project information and some of the main barriers for effective sharing of information done by DETR [1996c].

No.	Name	Description	Technical Issues	Organisational Issues
1.1	Design Information – General	Information traditionally exchanged as sketches physical models, drawings, specifications, calculation schedules, etc.	1. Forms of contract, delineation of responsibilities 2. Intellectual property rights 3. Professional indemnity insurance and such	1. Client awareness of the global project benefits versus locally higher costs of creating information so as to enable it to be shared easily and accurately 2. General commitment of client and other members of project management team to procedural disciplines in the generation and management of information
1.1.1	Geometry	Physical shape and the configuration of the elements or components of the building	1. Industry standards for component and connection details and classification schemas 2. IT standards for component data structures and for managing component connectivity and associativity rules	1. Ownership of standards, regulation of their implementation 2. Liability for failure in the event of otherwise compliant implementation of standards 3. Rules for managing non-standard implementations 4. Compliance with procedures
2.1	Management Information	Relates to the execution of the project and supports the project management process. It includes project plans and progress, information on resources, minutes of meetings, correspondence, etc	1. Industry classification frameworks 2. Project specific work breakdown structures 3. IT standards for the association of these attributes with components	1. Ownership 2. Compliance with procedures
3.1	Commercial Information	Procurement and payment processes	Generally single firm to single firm transactions, so of secondary importance in context of generalised project database	security

Table 3.1: Types of project information: the implications of sharing [Amended from DETR, 1996c]

3.2.1 Islands of automation and the need for information sharing

Today, sophisticated technologies, hardware and software, are being used to support the different requirements of each stage of the project life cycle [Alshawi, 1996]. These may include CAD systems, structural analysis/design packages, project management packages, etc. Although previous studies have shown that using these application packages can improve performance of the construction industry [Building Centre Trust, 1991; Wager & Scoins, 1984], the fragmentation in term of hardware and software still occur. Terk and Fenves [1995] highlighted that the current generation of computer tools is not designed to fit into the overall facility development process. As a consequence, this generates communication problems within the application packages used in the construction industry. This has resulted in "island of automation" within the industry, i.e. numerous applications are required to exchange and share information but no facilities available to do so [Alshawi, 1996].

On the other hand, the introduction of application packages on an *ad hoc* manner contribute very little to the performance of the overall project lifc cycle. The "islands of automation" exist in every stage of the project life cycle whereby relevant stages and their effect were not considered [Alshawi, 1996]. Several studies have been devoted towards finding the best solution for this problem. A study carried out in 1988 by the Construction Industry Institute (CII) has found that all respondents emphasised the importance of integration within their computer systems [Ibbs & Choi, 1989]. This study has also been supported by other researchers, Alshawi and Faraj [1995], Froese [1993], Sanvido *et al.* [1990], and Yamazaki [1992], who all stated that an integrated computer system that could facilitate information sharing and exchange is essential for an effective construction industry.

At the earlier stage of integration, the trend to integrate different computer systems was initiated by the need to manipulate and share information between a number of computer systems in a company [Hassan, 1997]. As computer technologies improve such as database management systems (DBMS), spreadsheets, computer-aided design (CAD) and knowledge-based systems (KBS), the development of the integrated computer systems also progressed through the combination of these technologies. For example, integrating CAD and spreadsheets [Birkbeck, 1991] to improve estimating function, DBMS and CAD systems for automating the generation of CAD drawings [Won & Klein, 1989], CAD and KBS for constructibility feedback for the layout and dimensioning of reinforced concrete structural elements during the preliminary design [Fischer, 1991], etc.. These developments were the catalyst for better and more effective developments in this field.

3.3 Approaches to integration

The construction industry is remaining highly fragmented compared with other industries such as the manufacturing industry. Fragmentation exists both within individual phases of the construction process, as well as across project phases from briefing, planning, design, construction, operation, maintenance and demolition [Howard *et al.*, 1989]. Due to this fact, the integration of design and construction has also been proposed and considered to be the optimal approach to successfully reducing fragmentation and eliminating some of the major problems in the construction industry

[Kangari & Sadri, 1996]. Many researchers such as Aouad [1996], Fischer *et al* [1994], Froese and Paulson [1994], and Howard [1989] have highlighted the issues of integration as a key factor for reducing fragmentation in the construction industry. Improved methods for generating, sharing, and maintaining project information is needed not only for reducing fragmentation but also increase competitiveness by bridging the gaps between and within project phases [Kartam, 1994].

Sadri and Kangari [1993] have suggested that to achieve full integration in the construction industry, information must be considered in two dimensions. First is horizontal integration across disciplines, and second is vertical integration, which is related to time and progress of the work throughout the construction process. Horizontal integration is the communication among professionals of different disciplines involved in one specific project phase. Vertical integration, on the other hand, means the exchange of information between professionals of different disciplines involved in different project phases. Kartam [1994] has stressed that both horizontal and vertical integration are essential for lessening prejudicial effects of industry fragmentation.

3.3.1 Definition of integration

"Integration" has become a watchword for much current research on computer tools for the construction industry [Rusell & Froese, 1995]. Therefore, it is not a surprise that information integration is attracting more thoughts and discussion in the construction circle than ever before [Aouad *et al.*, 1994]. From the construction view, Aouad *et al* [1994] have defined integration as the ability to share information or subsets of information between different actors/disciplines using a common model developed within a sound and reliable framework. Fischer [1989] define integration as a continuous interdisciplinary sharing of data, knowledge, and goals among the project participants. On the other hand, across the computer circles, Ibbs and Choi [1989] have defined integration as a technique used for a computer system to share a common database which can be accessed, used and updated by multiple applications or users.

Despite the clear definitions of the integration mentioned above, a thorough investigation of the key elements involved in developing an integrated system is necessary to achieve full integration. Over the years, a number of attempts have been made to develop such integrated systems. However, such attempts have been made without a solid theoretical foundation on which to base the framework design of the system [Abdalla, 1993]. Therefore, the key factor here is to define the right approaches to data exchange whereby integration can be fully achieved.

3.3.2 Approaches to data exchange

There are many aspects of integration that need to be addressed. One of the most important aspect is data exchange. Data can be exchanged in four ways, namely through a direct translator, a standard exchange format, a blackboard architecture, and a product model. Each of these techniques although applicable, imposes its own limitations [Hassan, 1997].

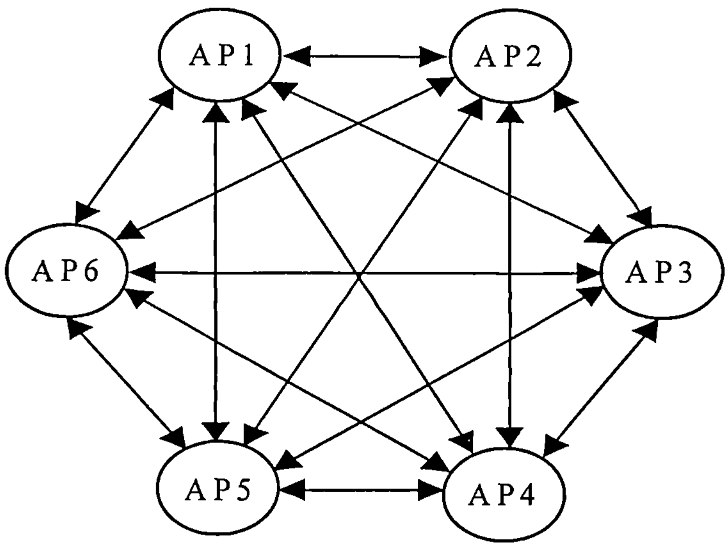


Figure 3.1: Direct translator approach

- *Direct translator approach* – the output data from one application program is directly translated into the format required for input to another application program as shown in Figure 3.1. The obvious disadvantage of this approach is the large number of translators that must be developed and maintained [Abdalla, 1993].

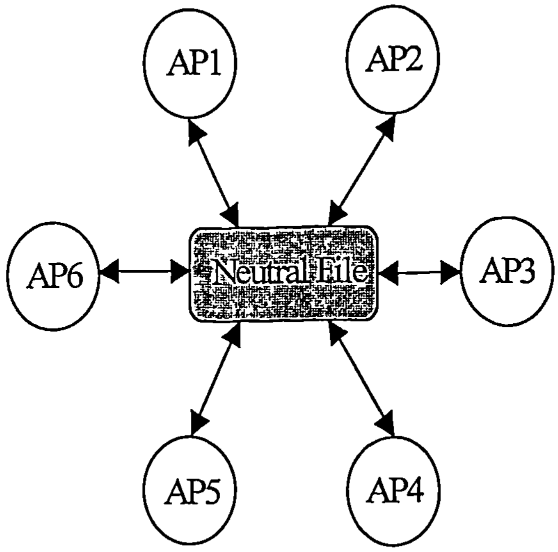


Figure 3.2: Standard exchange approach (neutral file)

- *Standard exchange format approach* – all application programs read input and write output through a single file using a standard exchange format that is understood by all the application programs (Figure 3.2). This standard exchange format file is commonly known as neutral file and widely used as the need for communication between various CAD systems. The "industry standard" format of DXF and IGES was accepted as a *de facto* for exchanging geometric information [Hassan, 1997]. The advantages of this approach such as uniformity to the system is introduced by using a single file and removing an existing application program will not effect the neutral file or other application program. However, using this approach a duplication of common data occurs whereby each application program only stores its own data which will results in consistency problems [Abdalla, 1993].

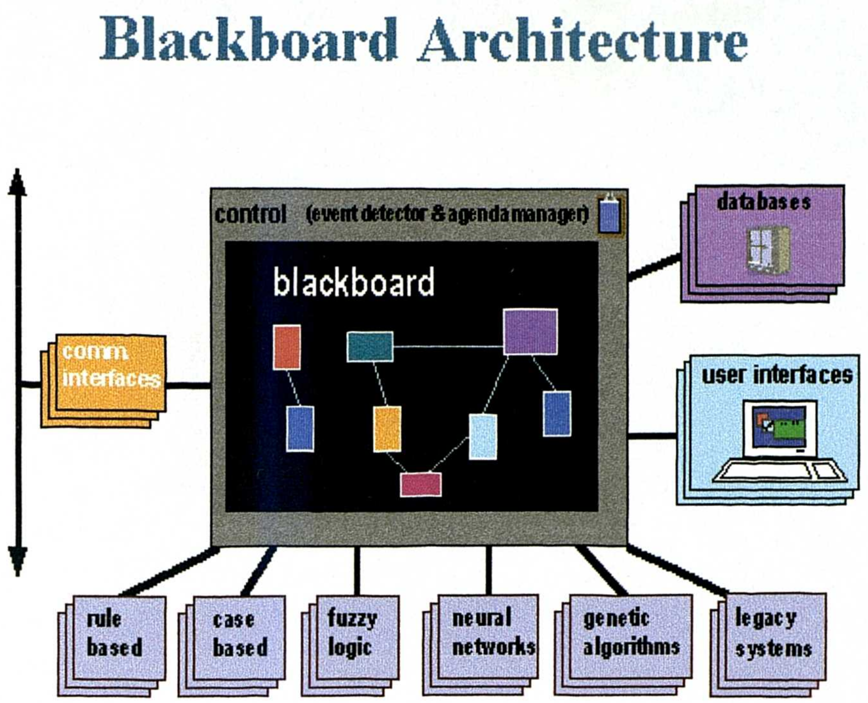


Figure 3.3: Blackboard architecture [Amended from Westinghouse, 1997]

- *Blackboard architecture approach* – A blackboard is a way of organising large amounts of data or knowledge in a manner, which allows different agents to act upon the data, almost concurrently [Branki & Bridges, 1993]. This is not a new approach but this concept was conceived by researchers in the field of Artificial Intelligence more than a decade ago [Westinghouse, 1997]. The use of this approach was to address issues of information sharing among multiple heterogeneous problem-solving agents whereby a group of experts gathers around a blackboard to collaboratively solve a complex problem. The blackboard is used as a central repository for all shared information. It also acts as a communication centre for its clients. In more precise terms, the blackboard may be thought of as a database, which represents the working memory of the problem solving system. The application of blackboard systems, though highly successful, have been one-of-a-kind, handcrafted efforts that require substantial investments in manpower in their design, development, and maintenance [Westinghouse, 1997]. For solving a complex problem in a specific domain, knowledge sources must be created to represent the necessary domain expertise. In addition, if the application requires interaction with external sources, the appropriate communications mechanisms must be put in place.

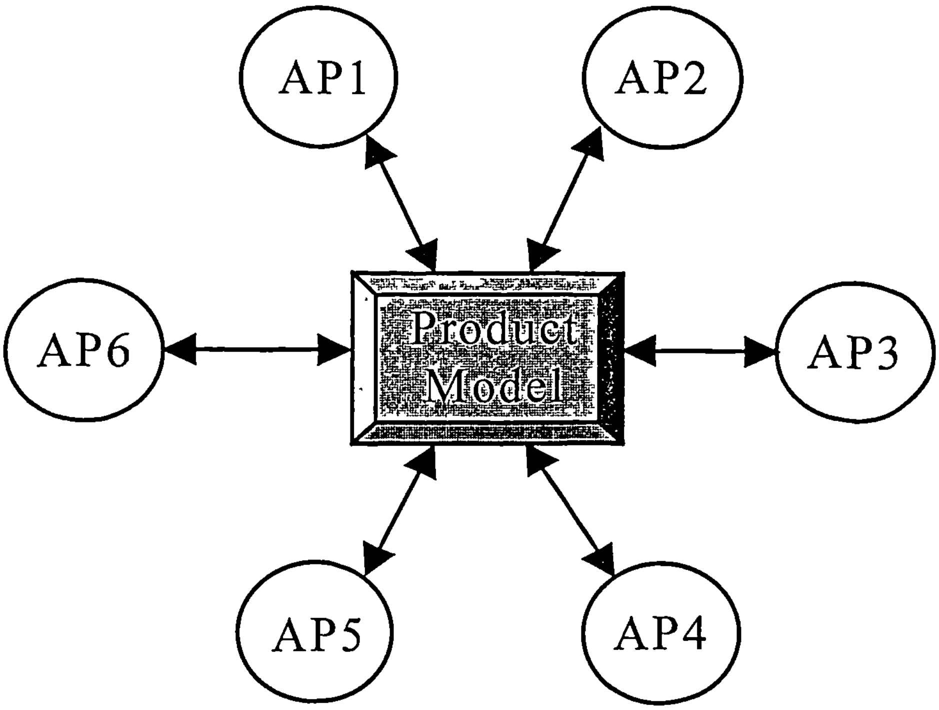


Figure 3.4: Product model approach

- *Product model approach* – This approach has been introduced to cater for all the problems mentioned above. For example, in the direct translator approach (traditional approach), large number of translators are needed as the number of applications increases. This will lead to data duplication, as each application requires its own representation of the product data. The standard exchange format was found to be inadequate to represent the project information since geometrical data alone such as line, circle, co-ordination, etc., cannot provide meaningful data in the form of an object which is important for other applications such as construction [Tah *et al.*, 1994; Kartam, 1994; Ewen & Alshawi, 1993]. The blackboard approach, on the other hand, has been recognised to supersede the traditional approach (direct translator), i.e. it allows for the fluent and transparent flow of information between the applications, it is suggested that such an approach still possesses the problems relating to expandability and maintainability [Underwood & Alshawi, 1997]. Therefore, the attempts to integrate additional applications into the environment are seen to be excessive and troublesome.

The use of a product model can reduce the problems stated above. It can reduce the required number of interfaces to integrate different construction applications and therefore facilitate information sharing, Figure 3.4. It eliminates the problem of multiple representations and transcription errors by having a single master model of the product [Alshawi, 1996]. Furthermore, the use of product model is seen as an enabling factor which, could provide richer representation of product data such as geometry, topology, relationship, etc. to completely define a component part or an assembly part for the purpose of design, analysis, construction, etc. [Hassan, 1997]. Detail description of product model approach is further discussed in Chapter 5.

3.4 Current approaches to integrated environments

As the industry witnesses more integrated systems, the need for better techniques of integration to support information sharing/exchange between professionals, departments, and entire organisations increases [Hassan, 1997]. Most integration research provides a technology push by developing a new technology and trying to push it onto an application [Betts *et al.*, 1995]. Betts *et al* [1995] stressed that a combination of technology push and strategy pull is likely to offer the best promise for successful implementation and application of an integrated environment. Many researchers see that there are an opportunities in construction for integration to allow fundamental improvements in performance [Alshawi, 1996; Yamazaki, 1992; Kangari & Sadri, 1996; Aouad *et al.*, 1994; Brandon & Betts, 1995]. Therefore, there is a current tendency throughout the world to view construction as a particular type of manufacturing process. This is shown that one of the major developments in manufacturing productivity and quality improvement from a technological viewpoint has been the development of Computer Integrated Manufacturing (CIM) [Brandon & Betts, 1995]. The ideal concept of integration in CIM which is currently being pursued to support the construction industry is called Computer Integrated Construction (CIC). The following sections briefly presents the concept of both CIM and how it led to the development of CIC as an approach to integrated environments.

3.4.1 Computer Integrated Manufacturing (CIM)

According to Shrensker [1990], CIM is the integration of the total manufacturing enterprise through the use of integrated systems and data communication coupled with new managerial philosophies that improve organisational and personnel efficiency. There is a question whether CIM is a concept or a technology, which is normally a common question. CIM is a concept, an environment, an objective, a strategy, and a technology [Vajpayee, 1995]. However, the aim of operating in a CIM environment cannot be achieved without modern technology.

The evolution of CIM started since mid-1970s and until 1980 it was merely a concept. However, in the 1980s, especially in the second half, CIM has expanded into the technology. There are several factors which have led to the development of CIM's concept and associated technologies, these are; the development of numerical control (nc) to control machine tools and other equipment, the advent and cost-effectiveness of computers, manufacturing challenges, such as global competition, high labour cost, product liability and demand for quality products, and the present of "islands of automation" of computer-based technologies [Vajpayee, 1995].

In the CIM approach, different modules (e.g. used by the design, manufacturing and shipping departments) are 'glued' together in an infrastructure which handles the flow of data between tools and let the different users efficiently engage in a collaborátive process [Gielingh, 1993; Geilingh & Suhm, 1993]. This concept gives similarity to the construction industry whereby many different disciplines and skills involved in designing and constructing a building interact either within or across company borders, using disciplinary and traditionally separated computerised tools [Augenbroe, 1995]. This concept is further expanded in the construction industry to develop the Computer Integrated Construction (CIC) duplicating the principles of successful CIM.

3.4.2 Computer Integrated Construction (CIC)

The concept of Computer Integrated Construction (CIC) emerged during the latter half of 1980s [Björk, 1992]. As stated in the earlier section, the presence of CIC has been largely influenced by the success of Computer Integrated Manufacturing (CIM) and also based on a fifteen-year-long research tradition in computer-aided building design [Vanier & Grabinsky, 1987]. CIC, in close conjunction with on-site automation and the use of industrial construction systems based on extensive pre-fabrication, has been proposed as one of the means the construction industry has of increasing its productivity and solving problems related to the quality of its products [Saarnivaara, 1990].

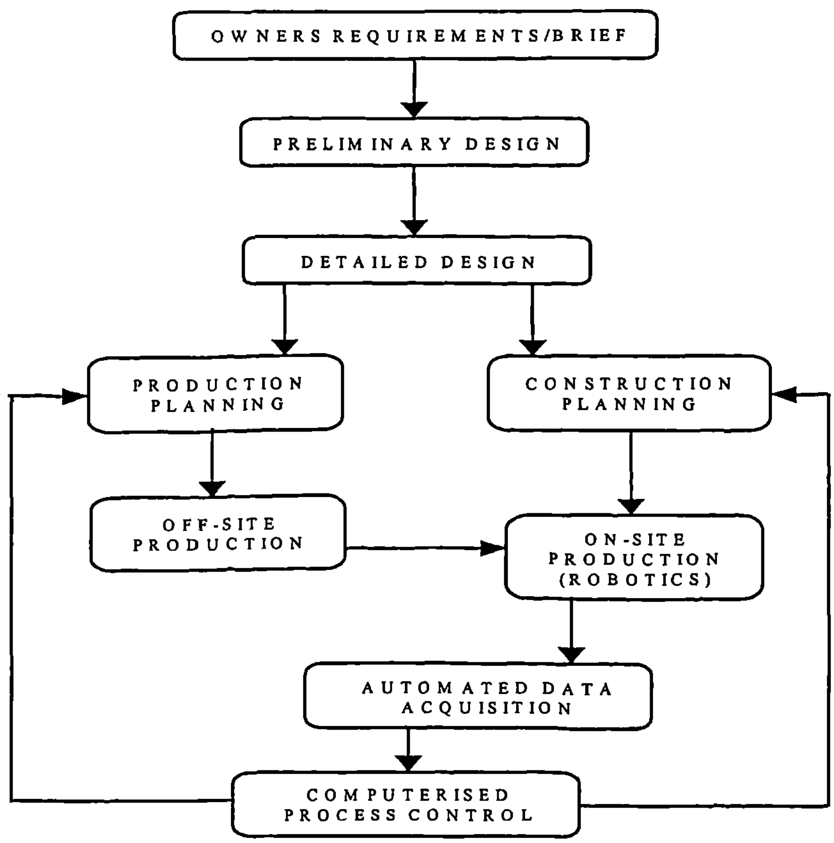


Figure 3.5: Computer Integrated Construction [Amended from Goldscmidt & Navon, 1996]

CIC has been defined as a business process that links the project participants in a facility project into a collaborative team through all phases of a project [Teicholz & Fischer, 1993]. Miyatake and Kangari [1993] have defined CIC as a strategy for linking existing and emerging technologies and people in order to optimise marketing, sales, accounting, planning, management, engineering, design, procurement and contracting, construction, operation and maintenance, and support functions. Goldscmidt and Navon [1996], describes CIC as the approach to the application of on-site real-time control which is an integral part of a comprehensive approach to automation of the construction process as shown in Figure 3.5. In CIC, all the activities are carried out with the aid of computers, starting from design, through construction planning, to the actual on-site construction. The construction on-site on the other hand is supported with various construction management software tools and is performed by robots. This description is also discussed elsewhere [Yamazaki, 1992; Eastman, 1994a; Navon, 1995].

Today, it is the efficient integration of information technology into application systems that is of greatest economic relevance to the construction industry [Choi & Ibbs, 1989]. The strategy for the automated flow of information is, therefore, of vital significance. In today's construction industry, information technology must be viewed as a potential resource. CIC is an emerging technology, and also an approach to assisting construction firm in responding to the difficult environment in which they currently operate. Miyatake and Kangari [1993] stressed that there is no standard formula for CIC, because the strategies for applying the concept is still being investigated. However, a strategy for implementing CIC should be formulated, and each company must define its own system. CIC consists of both the use of information technology in the different phases and tasks in construction, and the integration of these phases and tasks through the use of digitally stored data and data transfer [Björk, 1992]. CIC usually results from; an integrated information flow, the wide spread application of computers, and high levels of automation which represent the technological aspect of CIC [Miyatake & Kangari, 1993].

The approach used in developing CIC is based on the project model, which includes a physical description of the building and a description of the activities, needed to construct it [Goldscmidt & Navon, 1996]. The project model evolves with the project itself, being able to supply the needed data at each development stage for all the participants of the process. It has been widely recognised that in order to achieve the best performance, such integrated environment need to share information via a central core, i.e. a product model as discussed earlier [Alshawi & Che Wan Putra, 1995]. The product model is the core of any CIC environment whereby a library of objects is formed from which a symbolic project model can be built up.

3.4.3 Concurrent Engineering (CE)

The Computer Integrated Manufacturing (CIM) and Computer Integrated Construction (CIC) enable the widespread communication of product/project information among all product/project life cycle stages [Froese, 1995b]. Pisano and Wheelwright [1995] suggest, in a growing number of industries, it has become vital to excel at the simultaneous development of new products and new processes, rather than undertaking sequentially determined processes. It is therefore important for such integrated environment to share/exchange the information concurrently or simultaneously despite sequentially or by stages, which are normally, carried out in the product/project life cycle. Through this concept, the term concurrent engineering emerge as an engineering practice for giving the best results in developing an integrated environment.

Concurrent Engineering, which has emerged as an innovative and progressive paradigm, has become a fairly well known phrase in the design and manufacturing industry in late 1980s [Evbuomwan & Sivaloganathan, 1995]. The emergence of CE was driven principally by the need to cope with fierce global market competition. Concurrent Engineering (CE), *sometimes known as simultaneous engineering, or* parallel engineering has been defined as a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support [Winner *et al.*, 1988]. Broughton [1990] defines CE as an attempt to optimise the design of the product and manufacturing process to achieve reduce lead times and improved quality and cost by the integration of design and manufacturing activities and by maximising parallelism in working practices.

The purpose of concurrent engineering is to ensure that the decisions taken during the design of a product, results in a minimum overall cost during its life-cycle [Syan & Menon, 1994]. In other words, this means that all activities must start as soon as possible, to induce working in parallel, which additionally shortens the overall product development, process [Alshawi, 1996]. Concurrent engineering will also aid the members in a project team to be aware of the decisions that have been made during the early stages and all the subsequent stages of project life cycle as shown in Figure 3.6. For example, at any stage of the design processes, the designers can examine the impact of their design on client requirements, cost, construction, materials, environment, etc. The CE approach is totally different compared to the traditional approach as shown in Figure 3.7. In the traditional approach, each task of the project life cycle does not start until the previous one is completed. When the project gets to the construction stage, the contractors only have a little involvement. This will cause a long feedback loop between the various stages causing delays and may lead to expensive amendments.

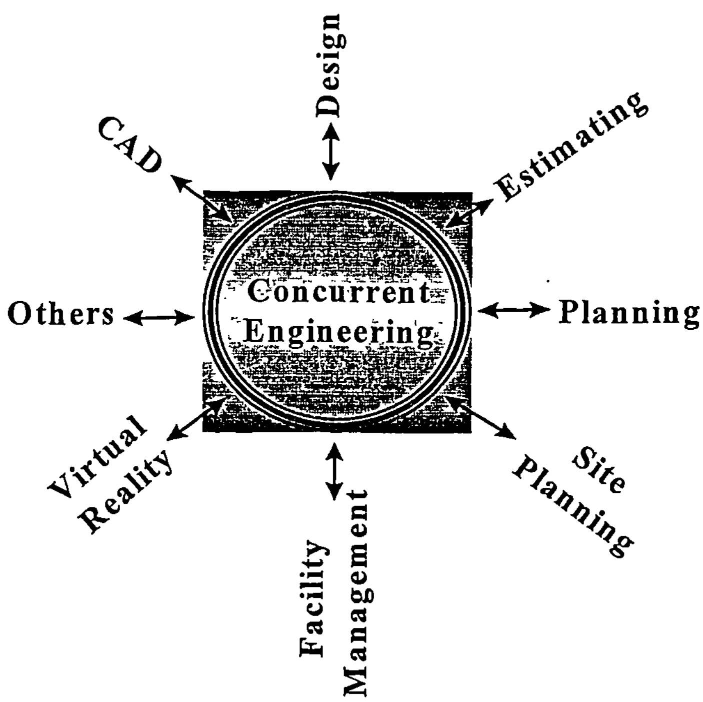


Figure 3.6: Concurrent Engineering approach to the project life cycle [Amended from Alshawi, 1996]

Concurrent Engineering allows all the disciplines in the construction process such as clients and suppliers, etc. to be simultaneously involved at every stage in the project life cycle whereby the information are shared/exchanged. This process could occur because in CE, each stage is independent from the others, i.e. one stage does not need to be completed before the next can start. Due to this fact, if implemented correctly, CE generates several benefits to the industry (construction/manufacturing). Some of the benefits are [Alshawi, 1996]; better communication and integration between participants, shorter project development lead-time, improved profitability, etc.

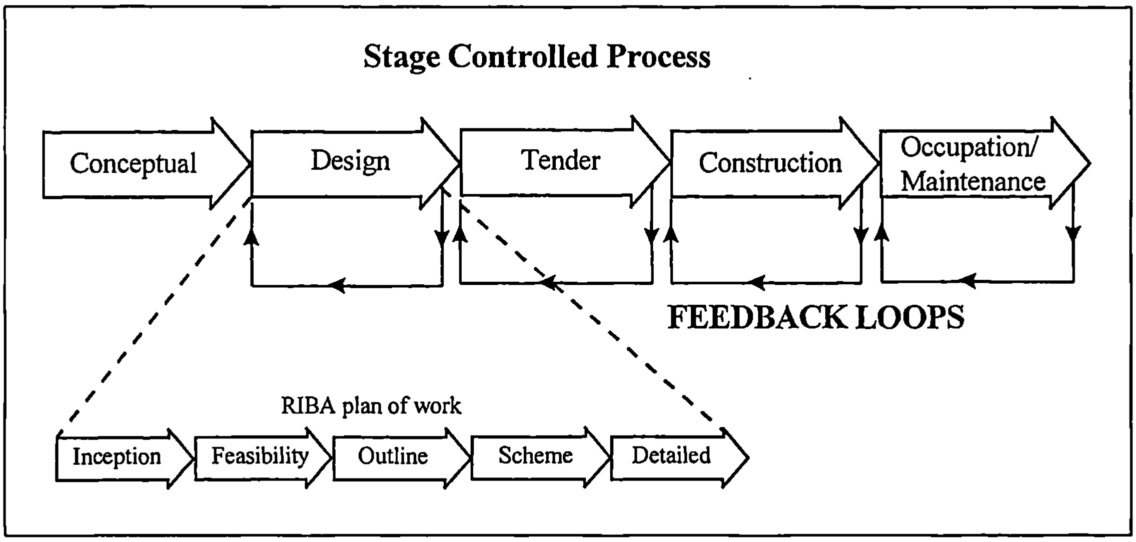


Figure 3.7: The traditional approach [Amended from Alshawi, 1996]

3.5 The needs for standards

The purpose of the development of an integrated environment such as CIC is to reduce/minimise the fragmentation problems occurring in the construction industry. Many researchers and industry leaders believe that the construction industry would benefit from greater integration or continuous interdisciplinary sharing of data, knowledge, and goals among project participants [Yamazaki, 1992; Kangari & Sadri, 1996]. These benefits could be gained by a means of standard communication which should support exchanging and sharing of the design information.

CIC requires data standards, or common information models through which a computer can exchange project information [Froese, 1995b]. The different approaches of integration, which have been mentioned earlier, can lead to the development of different standards of data exchange depending on the approach used. As mentioned earlier, a product model is the core of any CIC environment. Therefore, the main ingredient for the success of CIC activities is the development of a powerful and flexible product model [Ikeda *et al.*, 1996]. Many studies have shown that the information sharing and exchange need to be modelled and a standard is needed for the development of any CIC environment [Björk, 1992; Sanvido *et al.*, 1990; Yamazaki, 1992; Miyatake & Kangari, 1993]. Standards for data exchange will be further discussed in Chapter 4 while the development of product model in the integrated environment will be further discussed in Chapter 5.

3.6 Summary

The proliferation of procurement methods which is affecting the role of the professionals involved within the project life cycles has rendered the fragmentation of the construction process. The co-ordination of information and communication becomes difficult and results in breakdowns, misunderstanding and frequent litigation. Throughout the project life cycle, the information needs to be shared amongst certain functions within a stage. Therefore, the exchange and sharing the information between the various parties involved in the project life cycle stages becomes an important issue.

Recent studies have shown that computerisation can reduce the cost of the construction organisation and greater benefits can be gained by applying such technology to the management of the entire development process. On the other hand, the introduction of application packages within the construction industry on an *ad hoc* manner contribute very little to the performance of the overall project life cycle.

The integration of design and construction has been proposed and considered to be the optimal approach to successfully reducing fragmentation and eliminating some of the major problems in the construction industry. However, the key factor for the integration to be fully achieved is to define the right approaches to data exchange. Four different approaches have been highlighted such as direct translator, neutral file, blackboard and product model. The product model approach is seen to be the best approach for complex integrated environment.

The successful development of CIM has indirectly led to the development of CIC. Such integrated environments have been shown to maximise the efficiencies, productivity, project development lead-time, etc. However, it is suggested that if such integrated environment can share/exchange the information concurrently, the efficiencies and the productivity will be more maximise.

Chapter 4

Data Exchange and Standards

4.1 Introduction

Chapter 3 highlighted the needs of standardisation in an integrated environment. Problems such as, incompatibility of the hardware and software systems used within a company; the need to transfer data and information to other parties; and insufficient and inaccurate information resulting from the following traditional practices, give rise to the need of establishing an international standard for data exchange.

This chapter examines the issue of data exchange, with special reference to the construction industry. It explores the types and classifications of data exchange and looks into how they can affect the exchange of information between construction organisations, applications and individuals. Opportunities and problems are highlighted and international standards such as STEP and Application Protocols are also discussed. Finally, the latest development of Industry Foundation Classes (IFCs) by the International Alliance for Interoperability (IAI) is considered, including its claims to provide a basis for information sharing throughout the project life cycle, globally, across disciplines and technical applications. This investigation will be set against the backdrop of the need for an Integrated Environment such Computer Integrated Construction (CIC) and Concurrent Engineering (CE) which require a fast and stable data exchange between the organisations' applications.

4.2 Data exchange: Definition and classification

In the traditional approach to the project life cycle (which was discussed earlier in Chapter 2), the exchange of information from one stage to another is carried out manually. The emergence of computer technology and the introduction of integrated environment in the construction industry have increases the need for the automation of the exchange of information; enabling digital exchange as opposed to manual operations.

Digital data exchange can simply be defined as the transfer of information in digital form between computer systems [Watson & Boyle, 1993]. Generically, data exchange in construction is the process of transferring relevant or common information between different construction parties; with the aim of minimising data re-entry and duplication [Alshawi, 1996].

In construction, data exchange can take place between organisations, construction professionals and/or applications. Moreover, the needs for data exchange and the format by which data is represented can differ even within the same organisation. Therefore, in order to distinguish and understand the requirements and techniques required for efficient data exchange, the process can be classified into three levels, [Alshawi, 1996]:

- *Industry data exchange* – Industry data exchange is the exchange of data between organisations either in the same industry or in different industries. For example, the exchange of data between designers and contractors, contractors and suppliers, etc. Data can be exchanged through papers, disks or networks. For example, at the completion of a design, the design practice dispatches the construction documents to contractors for tender purposes. If both parties use computer technologies, which are compatible, disks or networks can be used to transfer this information.

- *Organisation data exchange* – Organisation data exchange is the exchange of data between the departments either on paper, disk or network. For example, in a design practice, the architect first produces the design drawings. These drawings are then, passed onto the structural engineer to carry out the structural design. The same drawings are also required by the services engineer in order to perform his/her design. These drawings are either exchanged using a paper format (printouts), in which case they have to be re-drawn for every upgrade, or using disks or networks if the design tools are compatible in terms of hardware and software.

- *Application data exchange* - The exchange of data occurs between two specific applications through the use of computer systems. In this type of data exchange, several approaches can be used i.e. package interfacing or central databases. Both approaches, allow real time data exchange. For example, the architect can share or exchange design information with other architects or structural engineer using CAD systems. This is done through industry standards such as DXF. Other examples are the use of standard database file using dbf format or using DDE (Dynamic Data Exchange) through Windows environment.

4.3 Data exchange: Problems and formats

Due to the complexity of the organisational structure, their operations and the vast amount of project information, the construction industry has encountered severe problems in replacing traditional paper-based information exchange with digital data exchange [Reed, 1988]. In order to overcome these problems, the industry has concentrated on proposing tailored solutions to specific types of data exchanged between industrial, organisational and application bodies described earlier.

At the basic level, paper-based data exchange is addressed, as currently most business operations are still paper-based. Whilst paper has certain tactile advantages, it is nevertheless expensive to handle and process. In the mixed media where papers and computers co-exist, the computer outputs still need to be re-entered into other computers. The compatibility may not be the problem since they are simply may not be connected electronically. Furthermore, even if they are linked in some fashion, they could have different format requirements in which the output format of one being different to the input format of the other. EDI (Electronic Data Interchange) has been developed as a solution to ease this problem. It is a standard for non-geometric data exchange.

The exchange of digital information can also be carried out by following a neutral format route allowing different systems to communicate with each other. In such a case, software vendors normally corporate to provide facilities that have the capability to write data in standard formats such as IGES, DXF, dbf etc. These standards, constitute what is known as the 'neutral file', Figure 4.1.

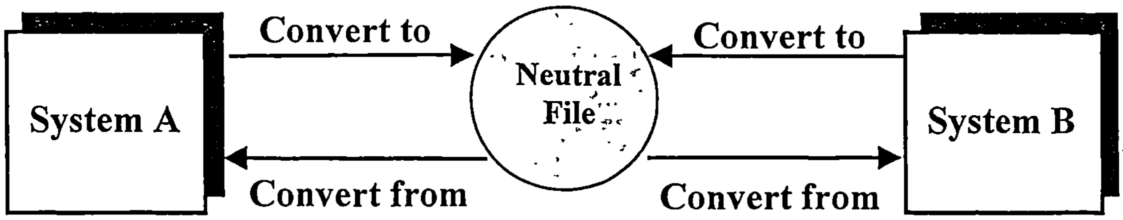


Figure 4.1: Neutral file format data transfer

4.3.1 EDI formats (Non-Geometric data)

EDI is the electronic transfer of business data from one independent computer system to another that has standards of data format agreed upon. In broad terms, EDI is essentially a method for exchanging documents between the computers of trading partners [O'Brien & Al-Soufi, 1993]. Unlike electronic mail or FAX, EDI involves data exchange between two applications. The data are transmitted by one application to the other in a structured computer processable format without the need for re-keying it

[Canright, 1987]. EDI has been specially aimed at automating trading data such as purchasing orders, acknowledgements, material releases, requests for quotations, etc.

Although, EDI originally focused on the exchange of common business operations such as purchasing, it has moved towards technical areas such as CAD data, product catalogues, etc. [Froese, 1994]. The development of EDI within the construction industry includes bill of quantities (BQ) and CAD data exchange. In order to implement EDI within an industry, there must be a well-defined standards and protocols for each type of message that need to be exchanged. Organisations such as UN/EDIFACT (world-wide organisation for establishing EDI), EDIBUILD (a Pan-European EDI user's group for the construction industry that looks at usage issues such as business requirements, implementation issues, liaison with other sectors, and trial use), MD5 (the technical body within the Western European EDIFACT Board that is responsible for message development and maintenance for the construction industry), CIAG (Construction Industry Action Group; a users group for EDI within the construction industry in North America), and PDIX (Process Data Exchange Institute), CIMIS (Common Industry Material Identification Standards; a user's group within CIAG and PIDX aimed at developing EDI standards for material identification) have been established to create these standards.

MD5 is currently developing a message set of bill of quantities, which is a listing of individual building elements, their quantities, and their costs [Froese, 1994]. The EDI message being developed will list individual bill items, which may contain sub-items, which can be attached to heading text and various indexes, and can be grouped at various levels. EDI messages have also been established for exchanging CAD data [Froese, 1994]. At present the actual CAD data elements themselves are too complex and heterogeneous in nature to be converted into EDI statements. Therefore, the approach used is to create EDI messages that will accompany, refer to, and describe CAD file such as DXF files, but not actually combined with the CAD file. The CIMIS group is working towards the establishment of global codes for identifying products. To date, they have set up their organisation, developed general data formats, and creating specific product format in areas such as piping.

EDICON has been created in the United Kingdom [Knowles, 1990]. The 'mission statement' of EDICON is to bring the benefits of EDI and electronic communication to the UK construction industry. EDICON includes quotation (including Price Sales Catalogue), purchase order (including Call-Off Order and Hire Order), order acknowledgement, invoice and payment. Since its establishment, EDICON has released standards for bills of quantities and materials descriptions [Sanders, 1988].

The benefits of EDI are obvious. The reduction of processing time for interparty transactions and the accuracy of the data is greatly improved. The human effort involved in processing transactions is reduced and there is an overall reduction in cost. The benefits at this level nevertheless improve in proportion to the number of trading partners who mutually adopt EDI.

4.3.2 IGES format (Geometric data)

In 1979, the Air Force ICAM (Integrated Computer Aided Manufacturing) project together with the General Electric Company and the Boeing Company have to develop a neutral data exchange file format. The outcome of this project, were the Initial Graphics Exchange Specifications (IGES) that were published in January 1980, and became an American National Standard (ANSI) in 1981.

IGES is strongly oriented towards the graphic capabilities of CAD systems. It defines a neutral representation of geometric and non-geometric data in terms of fundamental units of information called entities. Geometric entities represent the definition of 3-dimensional curves, surfaces, and solids, while non-geometric entities define the properties, aggregations, macros, and a number of graphic-oriented characteristics such as drawing definition, annotation, etc.

IGES is widely used and accepted among CAD/CAM users and vendors; it covers 2D co-ordinate, 3D wireframe, annotation, finite element, CSG (Constructive Solid Geometry) and B-Rep (Boundary Representation). Due to the large file size and the lack of functionality in IGES to support the emerging computer aided design, engineering, and manufacturing systems, other standards were also developed. Figure 4.2 shows the evolution of such development.

4.3.3 DXF format (Geometric data)

A 'de facto' standard such as DXF were developed to enable the exchange of graphical data among CAD packages. It was developed by Autodesk Inc., the developer of CAD software, AutoCAD. It has been introduced since the early release of AutoCAD. Later, in every new release, it has been extended to add more functions to it.

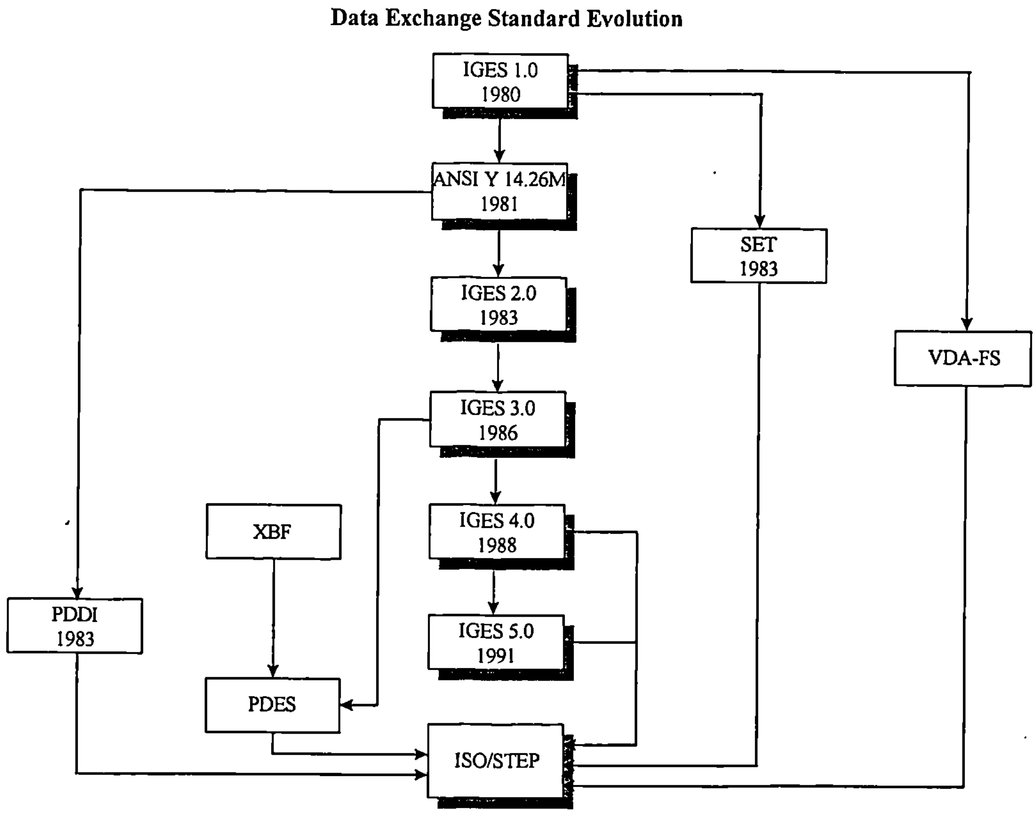


Figure 4.2: The evolution of data exchange format [Bloor & Owen, 1994]

DXF files are standard ASCII text files. It can be easily translated to the formats of other CAD systems or submitted to other programs for specialised analysis. DXF files can include CAD information either in 2D or 3D. The consistencies of the information/design, which is drawn within a CAD system, can be permanently maintained in a DXF file where other CAD systems can share or exchange the information.

4.3.4 STEP format (Geometric data)

STEP (STandard for the Exchange of Product model data) provides a representation of product information along with the necessary mechanisms and definitions to enable product data to be exchanged. The term product data denotes the totality of data elements which completely define the product for all applications over its expected life cycle for the purpose of design, analysis, manufacture, test and inspection [Smith, 1986]. The exchange of product data among different computer systems and environments is associated with the complete project life cycle, including design, manufacture and maintenance.

The purpose of STEP is to "create a standard that enables the capture of information comprising a computerised product model in a neutral form without loss of completeness and integrity, throughout the life cycle of the product" [McKay & Bloor, 1992].

The information generated about a product during these processes is used for many purposes. It may involve many computer systems, including some located in different organisations. To support such uses, organisations must be able to represent their product information in a common-interpretable form that remains complete and consistent when exchanged among different computer systems. Details on STEP and its architecture will be discussed further in section 4.4.1.

4.4 Data protocols

The proliferation in the number of data exchange standards offers no help to the users and vendors of CAD/CAM systems. Too many different construction application packages or systems have created "islands of automation", i.e. systems cannot communicate with each other due to incompatibility (even if they are connected).

In respect to the proliferation of exchange standards, a sub-committee SC4 within the ISO Technical Committee was formed in 1984 to propose solutions to this problem. The sub-committee concluded that a single international standard for product data exchange is needed [Zeid, 1991]. This standard must be complete, efficient, compatible with other standards and better than existing standards. To this end, the STEP concept was founded by the international community as one of the alternative for having a single standard of data exchange.

4.4.1 STEP protocols (APs)

STEP is an international effort aiming towards producing standards for exchanging high semantic-level product models that support technical information exchange and communication within industries. The purpose of STEP is to create a standard that would enable the capture of information comprising of a computerised product model in a neutral form (discussed earlier in section 4.3.1.4) without the loss of completeness and integrity, throughout the life cycle of the product.

4.4.1.1 STEP architecture

STEP has a three-schema architecture comprising of internal, external and conceptual as shown in Figure 4.3 [Yang, 1991]. The internal schema describes a physical representation of the data processing and storage. The external schema describes a particular usage of the data, i.e. it is dependent upon the application requirements and usage of the data. The conceptual schema is neutral to all application views, independent from any specific (external) or implementation (internal) views. It is divided into three levels, i.e. Generic Resources, Application Resources (AR) and Application Interpreted Model (AIM).

Generic resources are constructs sharable among multiple product types, domains, and life cycle phases, and free of context constraints. A formal language, EXPRESS

[ISO EXPRESS Committee, 1990], is used in defining the entities, types, functions, procedures, rules and references to support the applications in the conceptual schema. Application resources specialise generic constructs to establish specific relationships and constraints, which are sharable among multiple applications of a common application context. The Application Interpreted Model is neutral to all applications within a defined domain.

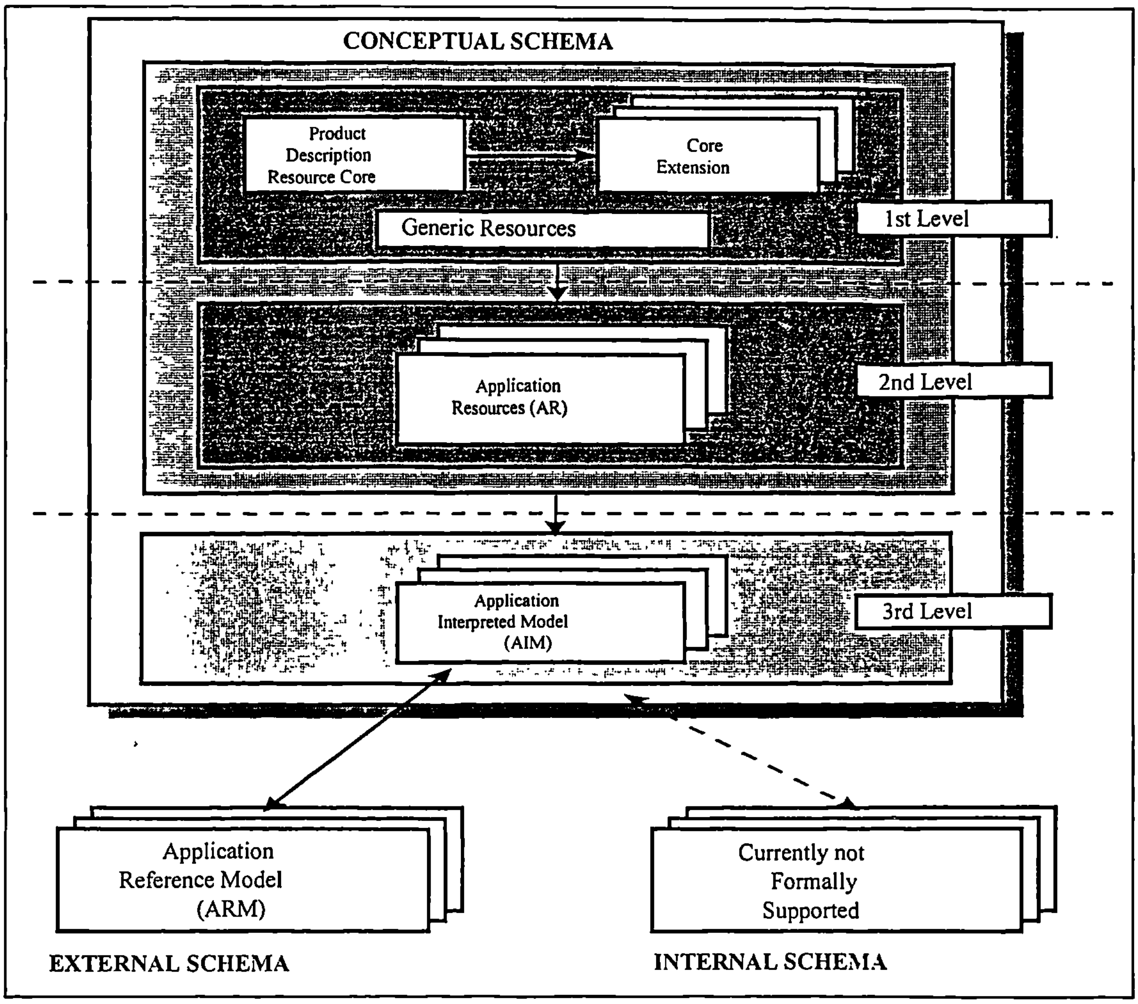


Figure 4.3: STEP Three-schema architecture [Yang, 1991]

4.4.1.2 STEP integrated resources

STEP Integrated Resources are either generic or application resources. They are assigned with the 40 and 100 series parts numbers of the ISO 10303. The 40 series parts contain generic resources while the 100 series parts contain application resources. Figure 4.4 illustrates the STEP Integrated Resources. In the current STEP Integrated Resources, it is divided into 5 categories. They are as follows:-

- **Product description resources**. These are generic resources that are required to specify the product data infrastructure. The product description resources are built upon a core structure, which consists of four conceptual constructs, i.e. product definition context, product definition, product property definition, and product property representation. The complete generic product description resources will be the combination of the core and its extensions, documented in other 40 series Parts. Currently, there are two extensions of the core structure: Part 43, the representation structure and Part 44, the product configuration structure.
- **Mathematical support resources**. These are generic resources that are required to support the specification of other generic resources using the mathematical definitions, such as geometry, topology, and measures. Geometry and topology are defined in Part 42 and measures in Part 41.
- **Presentation resources**. These are generic resources that are required to specify the visual presentation of displayable data, which includes: the organisation of the

presentation, elements in a presentation, display styles and properties. These generic resources are defined in Part 46.

- Management resources. These are generic resources that are required to specify product life cycle management data, such as approval of data and change of data. These resources are also defined in Part 41.
- Draughting resources. These are application resources that are required to specialise presentation data in the application context of draughting. These application resources are defined in Part 101.

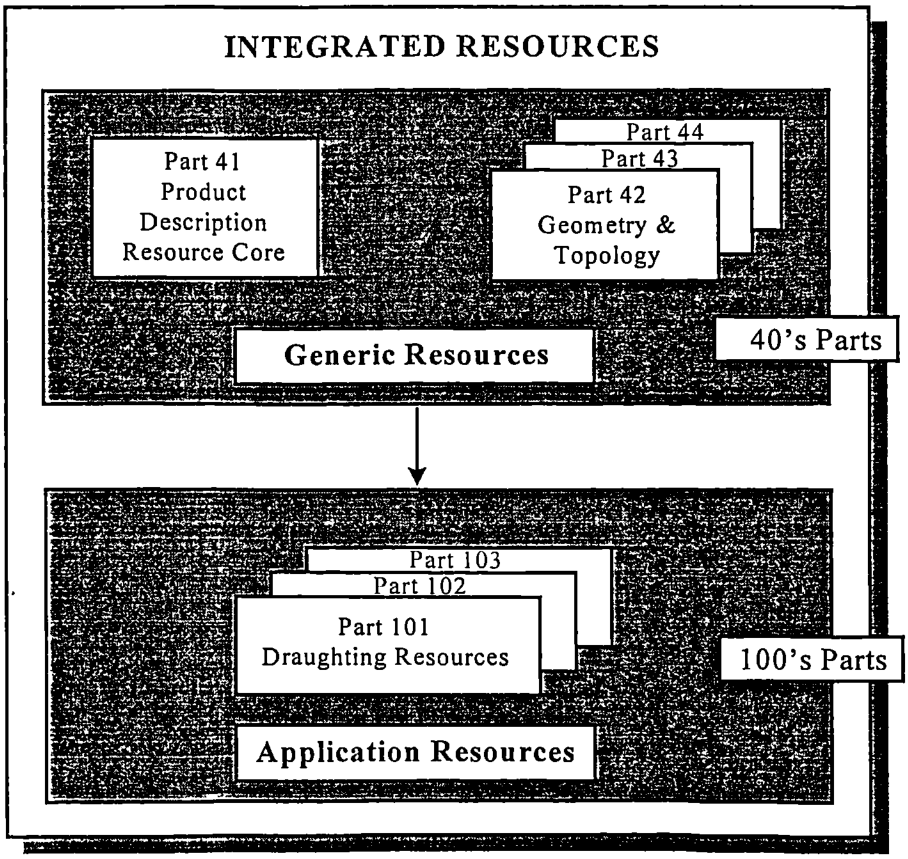


Figure 4.4: STEP Integrated Resources [Yang, 1991]

4.4.2 Application Protocols (APs)

STEP is built upon the key concepts of Product Models (PMs) and Application Protocols (APs). A Product Model is a formal information model describing how the totality of (engineering) information concerning a particular type of artefact can be coherently represented to facilitate information sharing [Watson, 1993]. A PM defines an agreed structure and definitions for the data, which relates to a type of artefact, which has to be shared. This specification is used to create a STEP data repository, which can then hold instances of the data to be shared. An AP is a standard that defines the context and scope for the use of product data and specifies the interpretation of the standardised integrated resources in that context to satisfy an industrial need [Palmer, 1993]. The STEP APs are designed to include the following five elements [Yang, 1991]:

- definitions of the application context, scope, and functional requirements;
- application reference model that describes the data requirements;
- application interpreted model that specifies the interpretation of the integrated resources construct to provide the required information;
- conformance requirements including set of test cases and their purposes; and
- usage guidelines that provide the information on employing the application protocol for exchange and sharing of product data.

The AP provides a complete and explicit statement of the product data description required for the specific need of a particular application domain. The AP defines the scope of the application domain. The Application Reference Model defines the product data information requirements and constraints for AP. The Application Interpreted Model describes a usage of the Generic Resources and Application Resources that satisfies the information requirements stated in the Application Reference Model [STEP part 1, 1991]. Currently, there are a number of STEP APs being developed, such as Building Construction Core Model (BCCM). The details of these APs are listed in Table 4.1.

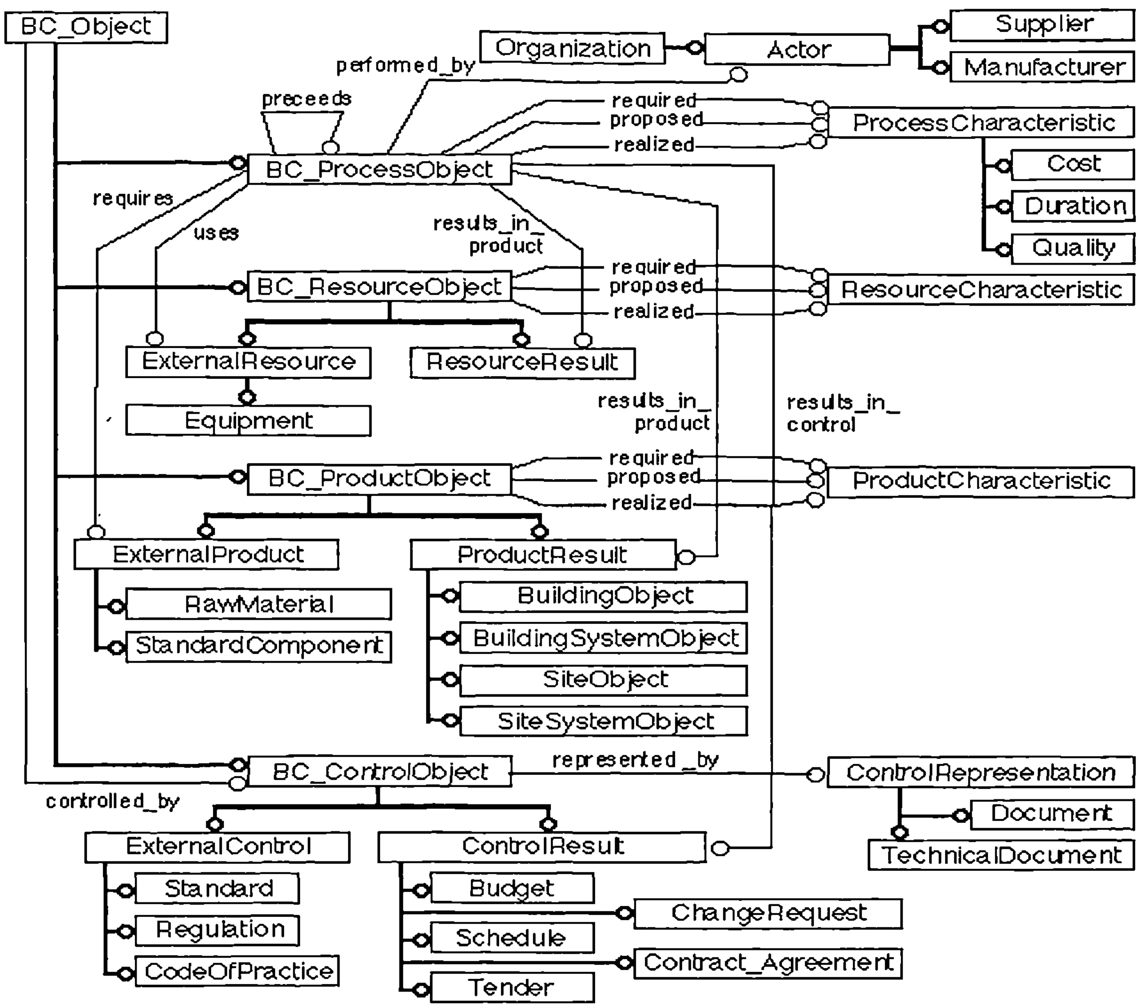


Figure 4.5: A portion of the STEP Building Construction Core Model [Froese, 1995]

Application Protocol Example	Background	Purposes of the Application Protocol	Scope and Applications
Building Construction Core Model (BCCM)	In May 1994, the Building Construction Core Model (BCCM) was proposed within the "Building Construction Application Protocol Planning Project" (APPP) submission and presented to the Project Management Advisory Group ISO TC184/SC4 [ISO TC184/SC4/WG3, 1994].	The purpose of the BCCM is to provide the overall definition of the entities and their interrelationships. These will enable different participants and applications software to use the definitions as a common basis for the exchange of project data and knowledge. The BCCM will provide a foundation set of data definitions for:- • an exchange between different software within different disciplines which makes available only that information which is necessary. • further development by more specific application definitions which would retain the integrity of the foundation set. The BCCM will propose an overall reference structure and entities for the common data requirements which will be available to the identified Family Groups for use directly within APs. Use of BCCM will ensure the consistency of approach required to support interoperability. It will also enable the transformation of data between different discipline views in different project life cycle stages. It allows the development a specific model for each discipline, using the semantics of that particular discipline, without being bothered by the semantics of other disciplines. For instance, architects will be able to use the decomposition, connectivity and semantics of a building most suited for architectural design whilst Contractors can use the decomposition, connectivity and semantics of the building most suited for realisation.	The scope of the model is to identify four major types of building construction objects. These are classified as: • Product Objects. Represent the systems and components of the constructed facility itself. • Process Objects. Represent the processes or actual construction effort on the project. • Resources Objects. Describe the resources used by the project such as material and equipment. • Control Objects. The items that control or constrain other project objects, such as contracts, budgets or standards. This model is intended as a reference for a number of more application-specific models, where more low-level details and attributes will be defined. A portion of the STEP BCCM is shown in Figure 4.5.

Table 4.1: Examples of STEP Application Protocols for construction

4.5 An international dimension to data exchange standards: The International Alliance for Interoperability (IAI)

The need to have global collaboration for data exchange that is free from imposed boundaries has brought the inception of an organisation called "The International Alliance for Interoperability (IAI)". It is an independent, open membership-by-subscription body; open to all companies in the construction industry. Its main objective is to promote the development and use of applications for the exchange and sharing of information to improve the efficiency and quality of building design, construction and maintenance. The IAI was established in September 1995 in the United States by 12 industrial leading companies who worked together to develop proof of concept prototypes demonstrating the viability of interoperability between AEC software applications. Currently, seven international Chapters have already been set up in North America, Germany, UK, Japan, Singapore, France, and Nordic countries. The individual Chapters have a Board of Management, represented on an International co-ordinated council [IAI, 1996].

Since it was first announced in April 1995, the IAI has attracted the interest of more than 150 companies and agencies. IAI seems determined to develop a "de facto standard" and focus on data exchange in construction. IAI will be committed to STEP compliance with an initial consideration for BCCM, AP230 (Steelwork), AP228 (HVAC) and AP225 (structural building elements using explicit shape representation). The IAI was formed to define, promote and publish the Industry Foundation Classes (IFCs) as a basis for information-sharing throughout a building's life cycle, across disciplines and technical applications. The following sections will discuss the IFCs, its benefits to industry, and its structure/model.

4.5.1 Industry Foundation Classes (IFCs)

One initiative of the IAI for interoperability is to set up international standards known as IFCs (Industry Foundation Classes). It is an association of leading A/E/C industry companies that includes manufacturers, design firms, construction companies, building owners, and software companies. Its purpose is to bring the benefits of interpretable software and intelligent building objects to all players in the building industry.

IFCs aims to provide a method for information sharing in the construction industry. They will supply a "common language" for defining a building project. Using object-oriented and component software technologies, IFC will provide customisable industry-based objects that encapsulate information about building elements as well as design, construction, and management concepts.

An important difference between IFCs and existing data exchange standards, both open and proprietary, is that IFCs will capture the relationships among the building elements. This makes IFCs objects act intelligently and will help capture the design intent at each stage of the construction process.

4.5.1.1 Benefits of IFCs

The benefits of IFCs will extend throughout a building life cycle, from the initial plans to demolition. The real power of interoperability, however, will be most visible when considering a late change in the design cycle. Changes will be made more quickly, easily, and accurately than are possible today, which will make the industry more responsive to changing user requirements.

This responsiveness will simplify and facilitate the use of new, fast-track methods for the design and construction of buildings. As clients require greater flexibility, lower costs, and shorter development cycles, information sharing through IFCs will become increasingly important.

4.5.1.2 Structure/Model of IFCs

The fundamental structure/model for the IFCs is aligned with that of the BCCM in STEP (which has been discussed in section 4.4.2). This structure includes four fundamental categories for entities, as follows:-

- **Products**. This part of the model includes all the physical elements in the project such as Building, Spaces, Walls, Doors, Windows, Equipment, etc.
- **Processes**. This part of the model captures information about the processes to design, construct, and manage the project.

- **Resources**. This part of the model defines the resources that will be consumed by the processes. This will include a general-purpose resource object that can be typed as labour, material, or equipment.
- **Controls**. This part of the model defines the constraints that need to be applied on the product, resources and processes. This will include architectural programme information, a design grid and other constraints.

4.5.1.3 IFCs object model

In the past, an AEC project was modelled by categorising its elements according to a primary functional role or as part of a system. This creates a significant limit to an AEC project because *many elements act in multiple roles and/or multiple systems.* However, in the IFCs object model, the model of elements, functional roles, and systems are defined separately so that an element can assume multiple roles and/or be a member of multiple systems [IAI, 1996].

Each structure of IFCs model (which has been discussed earlier in section 4.5.1.2) can be sub-classified in the IFCs object model as shown in Table 4.2.

Model	Core	Definition
Product Model	IfcProductObject	an object, manufactured or supplied for incorporation into an AEC project
	IfcSiteObject	an abstract supertype of each object type, deals with site
	IfcSite	represents zero or many buildings
	IfcSiteComplex	a collection of IfcSites
	IfcBuildingObject	an abstract which deals with the kind of things to construct and furnish the building
	IfcBuilding	information related to the building
	IfcBuildingComplex	a collection of IfcBuildings
	IfcBuildingStorey	space floor to floor, arch view of a container of space on more or less the same plane
	IfcElement	represents all the elements which define the building
	IfcSpaceElement	represents all types of space elements
	IfcAssembleElement	represents all elements which are assembled, constructed, or integrated into an assembly on the project site
	IfcManufacturedElement	represents the product elements which are pre-manufactured and placed, attached, or connected at the project site
Process Model	IfcProcessObject	represents the action or sequence of actions taking place in the building construction with the intent of acquiring or constructing an *IfcProductObject*
Resource Model	IfcResourceObject	item provided to assist in the process of designing, constructing, or managing an *IfcProject*
Control Model	IfcControlObject	a specification, regulation, constraint, or other requirement applied to a product or process whose requirements and provisions must be fulfilled

Table 4.2: IFC model overview [IAI, 1996]

4.6 Summary

The emergence of computer technology and the introduction of integrated environment in the construction industry have increase the need for the automation of the exchange of information; enabling digital exchange as opposed to manual operation. Data exchange can take place across several organisations, construction professional and/or applications. Data can be exchanged either in the form of non-geometric data such as EDI and geometric data such as IGES, DXF and STEP.

The international organisations with the collaboration of researchers around the globe have initiated the international standards which will be used in the computer integrated environment such as IGES and STEP aiming towards the development of a mechanism for the representation and exchange of a computerised model of a product in a neutral form. Due to the complexity of intelligently translating the geometric information between CAD packages using the current standards such as IGES, DXF, etc. and the emerging of new object-oriented software technologies, an initiative was taken to develop a new international standard namely the Industry Foundation Classes (IFCs) which is aimed towards providing intelligent objects for the construction industries that will dramatically increase the sharing of information throughout a building's life cycle, across disciplines and technical applications.

This chapter discussed the issue of international standard in the integrated environment. The issue of the development of International Standard Organisation (ISO), i.e. STEP is also discussed by stressing the current development of STEP within their Application Protocols (APs). Finally, latest development of Industry Foundation Classes (IFCs) by the International Alliance for Interoperability (IAI) committee was investigated. It is shown to have a mission to define, promote and publish IFCs as a basis for information sharing through the project life cycle, globally, across disciplines and technical applications.

As discussed earlier, the STEP protocols and IFCs are based on the development of a standard Product Model (PM) for data exchange. The next chapter will discuss the issue of Product Modelling and the integrated environment, their approaches and developments.

Chapter 5

Product Modelling and the Integrated Environment

5.1 Introduction

Modelling is a process of representing a perceived reality in a simplified way. Any model, by definition, leaves out some of the complexity of the perceived world [Carteret & Vidgen, 1995]. Modelling A/E/C information is a task that can be generally characterised by the complexity of the information and processes involved in construction industry [Van Leeuwen *et al*, 1995]. This complexity is contrived by the complex nature of the construction industry, including a great amount of inter-related aspects of information, many participants involved in the processes concerning the product, and different views on the product. Thus, the aim is to model this complex information in order to improve the representation of the information and to enhance the communication process.

This chapter discusses the 'Product Modelling' within the integrated environment. The earlier sections discussed the information modelling approaches by stressing the importance of each approach to design and construction. The following sections will continue the discussion on the definition, and the importance of Product Modelling. A brief description on the activity model using Structured Analysis and Design Technique (SADT) method and the data model using Express/Express-G method will be explained. Finally, the product models developed in an integrated environment such as RATAS, ICON, ATLAS, GenCOM, COMBINE, OSCON and WHISPER will be reviewed.

5.2 Information modelling approach

Information modelling or conceptual modelling consists of a collection of objects, their properties, their relationships with other objects, their memberships in classes, and their relationship constraints and cardinalities [Turner, 1992].

Atre [1988] defines an 'information modelling' as;

> "An inherent model of the entities with the properties representing them, together with relationships interconnecting the entities ....."

Alagic [1986; 1989] defines an 'information modelling' as;

> "A suitable representation of an application's environment...... An abstract representation of that environment relevant for the information requirement of the application"

There are three major approaches to information modelling in the construction industry [Björk, 1992a]. The first approach focuses on the flow of data between different information processing activities. This is normally shown in data flow diagrams. This approach is often used in projects with participants from the industry. In the second approach, the activities in a project and their causal relationships are modelled using graphical diagrams. Such diagrams are close to the traditional algorithmic way of defining computer systems. The third approach focuses on modelling the structure of the information describing the products, processes, resources and other elements of the construction process using conceptual modelling techniques. This approach has been particularly popular in product model research. It aims at determining the data structures used for storing construction data in databases. Such approaches are normally known as follows:-

- Activity Model
- Data Model
- Product Model

The following sections describe in details each of the above approaches.

5.2.1 Activity model

Activity model or process model technique has been used in a number of projects for analysing the information flow between different information processing activities in the design and construction [Björk, 1992a] such as design, procurement, estimating, planning, etc. and data and materials flow between these processes. The modelling approach helps in identifying the various processes involved and the data they require. This can be represented in an interaction model which shows what data is required for a certain process [Aouad, 1992].

The Data Flow Diagram (DFD) is a technique for activity modelling which shows the input and output of the concerned process and also a breakdown of each of the main processes [De Marco, 1978]. However, more structured approach has been introduced such as the SADT (Structured Analysis and Design Technique) method using IDEF∅. This method has more advantages over the DFD such as the ability to show the control and mechanism for the processes [Marca & McGawn, 1988] (details of this technique will be discussed further in section 5.4.1). On the other hand, the emergence of new object-oriented analysis methodologies has introduced a new activity modelling techniques such as Object Flow Diagram (OFD) in James Martin object-oriented methodology [Martin, 1993].

The advantage of activity modelling in the construction industry is to simplify the understanding of the involved processes and the development of their subsequent applications. However, activity modelling cannot be used as a means to translate conceptual models into physical databases [Aouad, 1992].

5.2.2 Data model

Data model is a set of concepts that can be used to describe the structure and operations of a database [Elmasri & Navathe, 1989]. The idea of a data model is to define the behaviour to be included in a database in response to the complexity of constructing large database for organisations as proposed by Sibley [1974]. As described by Sibley [1974], a data model is a model of the real world concepts and relations that are to be encoded into a database. A data model also provides the basic structuring mechanisms for describing the data, relationships and constraints of the information stored in any information system [Björk, 1992a]. Using a specific data model, conceptual models can be built.

The basic concept used in almost all data models is the object or entity [Augenbroe, 1992]. An object is a set of interrelated data about some 'thing' in the modelling domain [Björk, 1992a]. Object can be interrelated via relationships. For example, a door object may have the relationship with the wall into which the door is fitted. Some data models also contain the concept of 'attributes'. An attribute is a piece of information at the atomic level - that is, in a form such that it cannot be subdivided into meaningful component pieces [Carteret & Vidgen, 1995]. An attribute can be used to specify a property related to an object (e.g. a door height). In some data models, attributes can only be primitives data types (e.g. integer, real numbers, etc.).

Various conceptual data models have been developed that are rich enough to capture the relations and issues in some application areas, in a formal language that has a clear logic or mathematical basis [Eastman, 1992]. Traditionally, the tool used for data model is the Entity Relationship Diagram (ERD) [Chen, 1976]. However, STEP developers used other tools such as NIAM [Nijssen & Halpin, 1989], IDEF1X [Appleton, 1986] and EXPRESS [Schenck, 1989].

5.2.3 Product model

The STEP standard (discussed earlier in Chapter 4) is closely linked to the concept of Product Models. A Product Model can be thought of as a set of entities which together forms a system [Wright *et al*, 1992].

Geilingh [1990] defines a Product Model as;

> "A Product Model contains 'complete' and non-ambiguous information about a product. Complete in the sense that all information required for a specific application, if available, will be specified as an integral part of the product-model or can be derived from it".

Watson [1993] defines a Product Model as;

> "A Product Model is a formal information model of how the totality of (engineering) information concerning a particular type of artefact can be coherently represented to facilitate information sharing".

A Product Model is also the database model supporting the design, fabrication, operating and other uses of some type of product [Eastman, 1992].

Since the late eighties, the concept of Product Model has been widely recognised as the key to data integration [Turner, 1988; Reed, 1988] whereby the Product Model is designed as the central core where the information can be shared. This statement is supported by Augenbroe [1992], who stated that a Product Model could conceptually be regarded as the central core to which all clients are related and to which all clients exchanged their data. Particularly, a Product Model of a building comprises all data that is needed for a complete description of the product in its different stages and hence supports the extraction of all kinds of different views of the product [Augenbroe, 1992].

5.3 Definition and importance of Product Modelling

Product Modelling is an attempt to define the logical process of creating and generating of information for particular product types [Cornick & Wix, 1988]. Such a product type may be of any aspect of a building such as a spatial system, services system, cladding system or even the whole total building system. The purpose of Product Modelling includes the representation of product data throughout the life cycle of the product [Van Leeuwen *et al*, 1995]. These involve modelling multiple aspects of the product, such as aspects of design, cost, planning, etc. This aspect can in turn be seen from different viewpoints of parties involved at every stage in the product life cycle.

Product Modelling is considered to be an established concept not only for semantically based data exchange, but also for the specification of models, dealing with specific application requirements [Junge & Liebich, 1995]. Many researchers [Spur *et al*, 1986; Gielingh, 1989; Björk & Penttilä, 1989; Cornick & Wix, 1988; Greening, 1995; Froese, 1995b] have already highlighted the importance of Product Modelling approach in the integrated environment.

As discussed in Chapter 3, the product models can reduce the required number of interfaces to integrate different construction applications and facilitate data sharing. Therefore, it will avoid the problem of multiple representations and transcription errors by having a single master model of the product. Furthermore, the Product Modelling approach enables applications to access the data in the product model and/or write or update existing data, thus maintaining the consistency of product data held in the product model [Alshawi, 1996].

Through the product modelling approach, project data can be effectively maintained, managed, and manipulated. Updated information can be instantly accessed by different professionals, data can be exchanged between different parties and the impact of one party's decision on the other can be examined. In conclusion, this approach can provide the necessary vehicle which is required to bring about integration in the construction industry [Alshawi, 1996].

5.4 Information modelling techniques

The following sections will discuss the techniques, which have been used in this study.

5.4.1 The SADT method

SADT (Structured Analysis and Design Technique) is an activity modelling technique, which was originally developed by Ross [Ross, 1977; Ross and Schoman, 1977]. In his introduction to SADT, Ross stresses that with the use of SADT, human thought can be expressed concisely and clearly [Marca & McGawn, 1988]. Since the emergence of SADT, it has been widely used in the development of Computer Aided Engineering (CAE) application systems. The US Air Force Integrated Computer Aided Manufacturing (ICAM), adopted the technique and resulted in the IDEF∅ (ICAM Definition Method Zero), which is now been used by the STEP initiative to describe activity based models. An important feature of the SADT is its time dependency, i.e. activities are sequenced according to their time and extension. This criterion makes SADT different from other structured techniques such as Data Flow Diagrams (DFD).

SADT is an activity based modelling technique, which adopts top-down approach whereby high level activities are decomposed into low level activity exposing more and more, details of the system being analysed. The central feature of SADT is the "SA box" (Structured Analysis box), and its concept of "decomposition" [Marca & McGawn

1988]. The SA box is a graphical representation of an activity along with its main four types of data flows; the main activity's input, the activity's constraints and controls, tools required to perform the activity, and the activity's main outputs. In a context diagram, the purpose of an activity can be clearly outlined by expressing what result must be produced, what input data are available, what tools are to be involved, and the constraint conditions that must be considered. The context diagram is then decomposed into many sub-activities, the sum of which performs the activity described by the context diagram. This is shown in Figure 5.1. In each SADT diagram, activity is represented by boxes, and the relationship and flow of data between activities by lines and arrows.

In each activity diagram, there may be four different types of data. These are:

Input: Use to represent the information or material, which will be transformed by the activity as shown by an arrow at the left of an activity box, and marked I_n.

Output: Use to represent the result of the transformation as shown by an arrow at the right of an activity box, marked O_n.

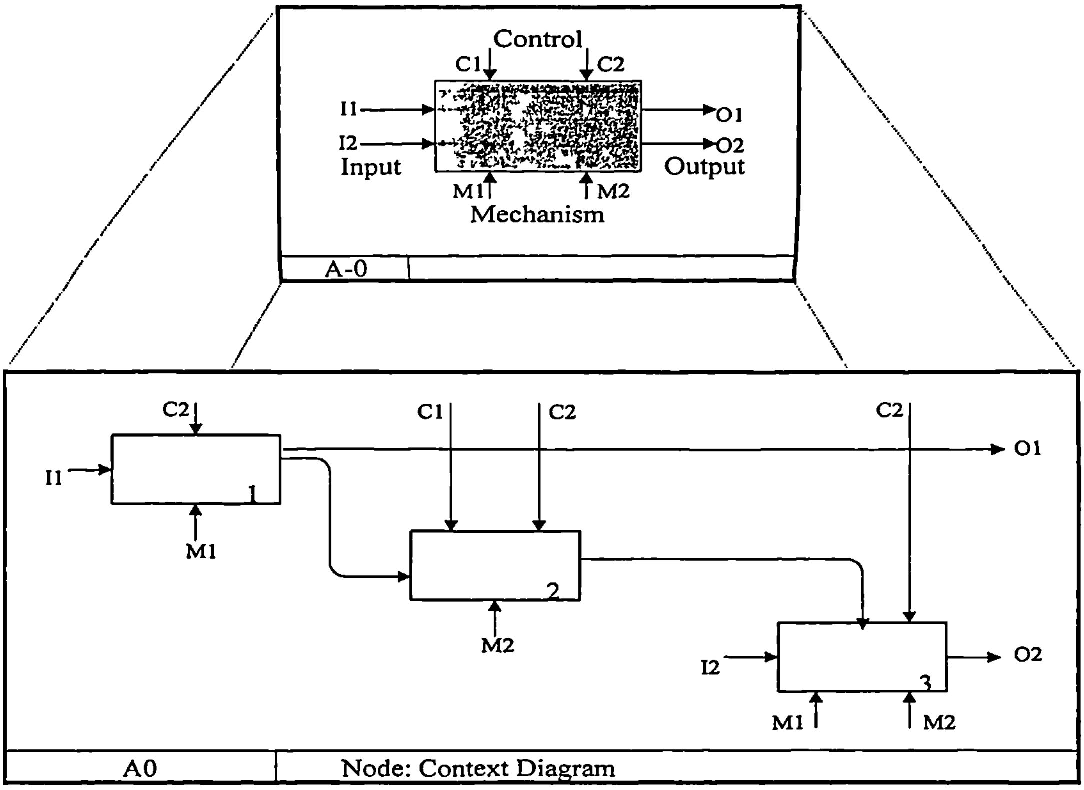


Figure 5.1: The Context Diagram and its decomposition [Grabowski, 1991; Marca, 1988]

Control: Use to represent the information which will influence the activity function as shown by an arrow at the top of an activity box and marked C_n.

Mechanism: Use to represent the tool or tools that support the activity or the human being who realises the activity as shown by an arrow at the bottom of an activity box, marked M_n.

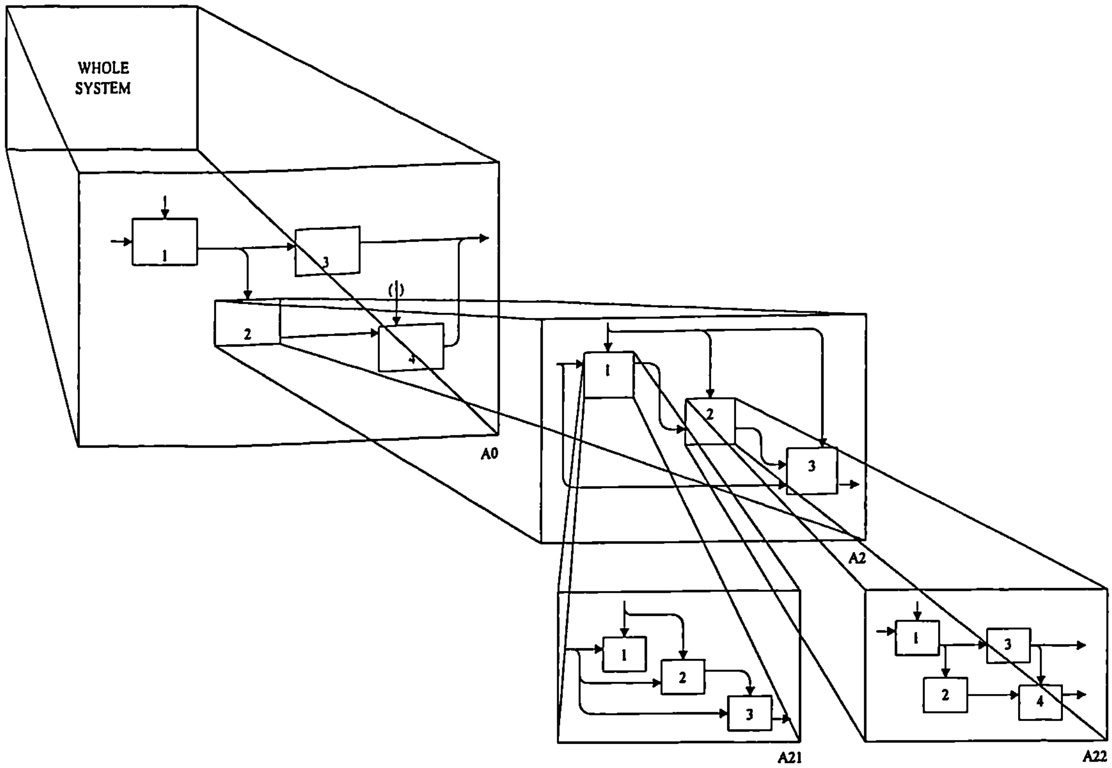


Figure 5.2: The hierarchical structure of SADT diagram [Grabowski, 1991; Marca, 1988]

The activity in a SADT diagram can be interpreted as: under the *control* conditions, the *inputs* are transformed into the *outputs* using the *mechanisms.*

Each activity box may be subdivided repeatedly until the required level of detail is reached. Usually three or four levels will be needed. Figure 5.2 shows the hierarchical structure of the subdivision. A fully developed SADT model graphically captures the characteristics of a system and helps a person to determine the activities in any system together with their relationships.

5.4.2 The EXPRESS and EXPRESS-G method

A number of tools are available for developing and defining conceptual models. Some of these are graphical and very useful for early sketching work and presentations. Rather than using traditional structured methods of analysis, this study opts to implement the graphical counterpart of EXPRESS, i.e. EXPRESS-G [CEN, 1991] which is a technique to produce data models. This is mainly due to its mandatory use in the STEP product modelling standardisation effort. The following sections will discuss briefly the overview of EXPRESS and EXPRESS-G method.

5.4.2.1 EXPRESS

EXPRESS is an information modelling which consists of computer-readable data definition language [Schenck, 1989]. EXPRESS is a large language, with extensive syntactic forms for defining complex data structures and functions [Eastman, 1994]. EXPRESS not only adds exactness to the data model definition, but it also adds a number of concepts which cannot be modelled in standard graphical tools, like rules, constraints and methods. These concepts render the language of an object-oriented flavoured [Augenbroe, 1992]. This object-oriented flavoured information model specification language was initially developed in order to enable the writing of formal information models describing mechanical products. It is one of the technologies that have been developed as part of the STEP standard for product model data exchange

[Schenck & Wilson 1994]. It has been developed over a number of years and has undergone significant change, even between the Draft International Standard version and the final International Standard version.

EXPRESS describes a domain in terms of the objects (called ENTITIES in EXPRESS) in that domain, their characterising attributes and the relationships that exist between objects. The relationships may be structurally constrained so that they may be simple and aggregated. Support is provided for derived attributes and optional attributes and there is a mechanism for capturing existence dependency between objects.

Entities can be further specialised in EXPRESS using the subtype clause, which allows the inheritance of the data structures of a supertype to its subtypes. It is possible to redefine the attribute of a supertype in a subtype, provided that the definition in the subtype is narrower than in the supertype. In addition, it is possible to constrain the information with the help of rules defined using EXPRESS syntax such as cardinality rules. EXPRESS also allows the definition of operations on the attributes in the form of functions or procedures.

5.4.2.2 EXPRESS-G

EXPRESS-G is the graphical representation of EXPRESS, using graphical symbols to form a diagram [Eastman, 1994]. It can express only a subset of the full language, including the notions of entity, type, relationship and cardinality. It also separately supports the notion of schema. It does not provide any support for the rules and rule mechanisms. In EXPRESS-G, entities are represented by rectangles, with the name of the entity indicated inside the rectangle. Predefined simple data types, such as integer, string, etc. are symbolised by rectangles with a double vertical line at the right end of the rectangle [Björk, 1992c].

Lines represent relationships between entities. Relationships, which are modelled as optional attributes (cardinality zero or one) are symbolised by dashed lines. All other relationships are symbolised by normal lines. In these, the circle is attached to the entity which functions as the data type of the explicit attribute of the other entity. In some cases, the inverse relationship may be indicated. Aggregate data types in relationships may be indicated by abbreviations such as S or L followed by the cardinalities. Thick lines are used to symbolise supertype-subtype relationships. A small terminal circle on the line indicates the subtype end of such relationship. The schema in Figure 5.3 illustrates the use of EXPRESS-G.

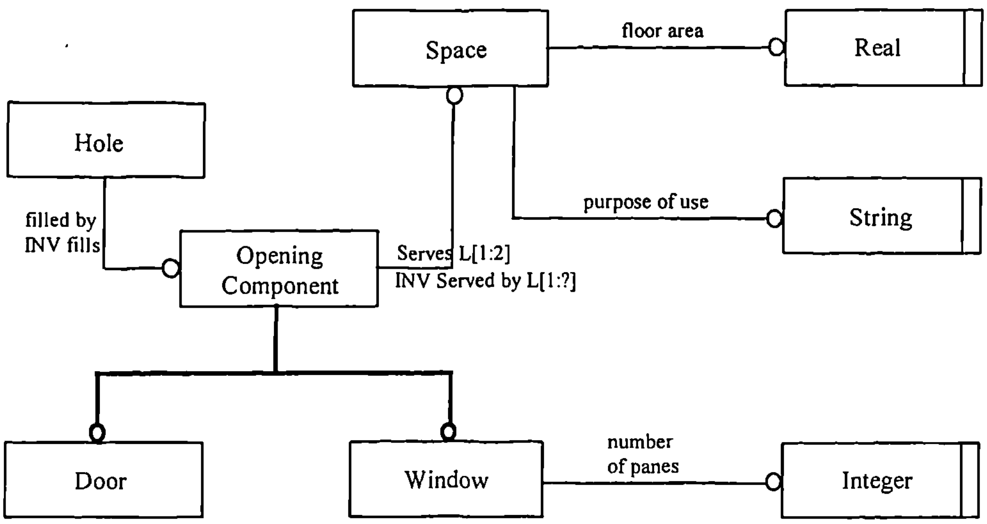


Figure 5.3: A small example schema illustrating the symbols used in EXPRESS-G [Björk, 1992c]

The above example schema in Figure 5.3 shows that there are eight entities. Three of these are simple or terminal data types, i.e. integer, real, and string. The other five are more complex data types. The *Door* and *Window* entities are both subtypes of the supertype *Opening Component.* A window has an attribute *number of panes*, which is represented by a simple data type. Opening components are related to both *Spaces* and *Holes.* A hole may or may not be filled by an opening component. Thus a hole entity has an explicit attribute filled by, the data type of which is an Opening.

An opening serves one or two spaces. This means that the opening component entity has an aggregated attribute serve, which is of data type space. The L is an abbreviated form of SET and the numbers within the brackets indicate the lower and upper bounds on the cardinality. Since this is a many-to-many relationship, we also need the inverse relationship, indicated by INV, which specifies that one to many openings may serve each space. Details of this technique will be further presented in Chapter 8.

5.5 Product Models in an integrated environments

Numerous Building Product Models have been defined in research projects, some of which are being used in practice [Van Leeuwen *et al*, 1995]. Early research projects resulted in models that were either generic for a type of building or specific for a particular domain. RATAS project for example, presents a generic information hierarchy for buildings [Björk, 1989]. More specific models represent a subsets of the life cycle of a building, for instance in the domain of HVAC engineering or structural engineering. However, current research is moving into the direction of complete life cycle models, mostly defining a generic or kernel model to which view-specific models are related, for instance in the COMBINE project [Augenbroe, 1993].

The following sub-sections will review the previous and current developed research on Product Models in an integrated environment such as RATAS, ICON, ATLAS, GenCOM, COMBINE, OSCON and WISPER.

5.5.1 RATAS

The RATAS-project is an effort to develop a national Finnish system for computer aided design in the construction industry [Enkovaara et al, 1988]. The RATAS building product model was developed as part of the project to achieve a national building product data model standard. The model is described in [Björk, 1989] and [Björk & Penttilä, 1989].

The RATAS model describes a building using objects (entities), and relations between objects. The object description of a building is made up of objects and of a network of relations between these objects. Together these constitute a product model of the building. There are two types of relations involved, the "part-of" and the "connect-to" relations. "Abstraction hierarchy" is used in the modelling work, to make a sub-division of the building into meaningful systems and parts. An "Abstraction

Hierarchy" is a tool which helps designers to deal with the subdivision of a building into meaningful systems and parts and with the internal relations between such objects [Eastman, 1978].

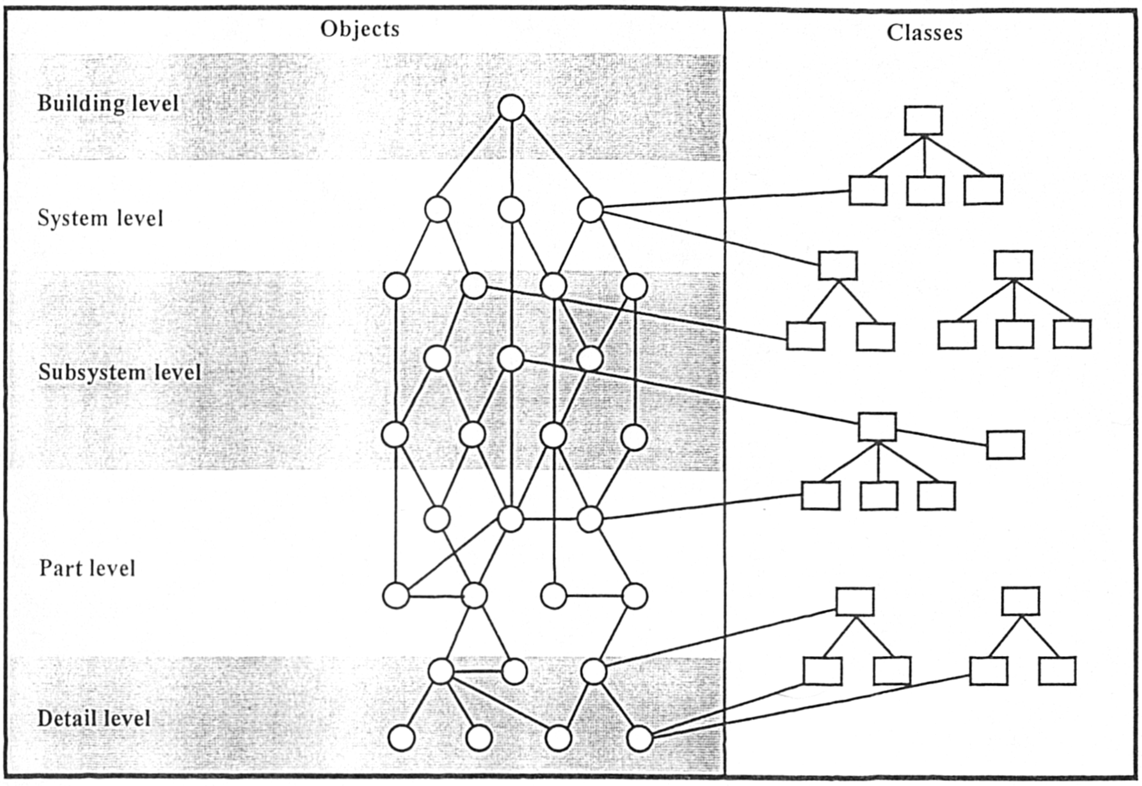


Figure 5.4: An illustration of the levels used in the RATAS system [Björk & Penttilä, 1989]

The model distinguishes five basic abstraction levels:

- *building level* - contains attribute data about the site, building type, etc.,
- *system level* - contains general information about the building systems, e.g. spaces, structural type, etc.,
- *subsystem level* - subdivides the systems into functional parts,

- *part level* - contains physical objects such as building elements or technical devices, and
- *detail level* - contains detail information about an element such as a window with its constituent parts.

An advantage of the RATAS model is that it has been developed in close contact with the construction industry. However, RATAS is also theoretically well founded. The "abstraction hierarchy" of the RATAS model is a composition level order, where things in lower level are parts of things in higher levels. However, the concept of level is not explicitly defined. There is a risk that objects in higher levels are regarded as more "abstract" and objects in lower levels more "physical".

5.5.2 ICON

ICON (Information/Integration for CONstruction) is a research project undertaken at the University of Salford, UK, aiming at investigating the feasibility of establishing a framework for integrated information systems in the construction industry [Aouad *et al*, 1994]. It uses state of the art methods and information modelling techniques such as Information Engineering Method (IEM), Object Oriented Analysis and Design, and Computer Aided Software/Systems Engineering (CASE). Tools.

ICON uses the method which involves separating the construction into several object oriented models, each describing the information needed to support a single well-defined activity [Cooper *et al*, 1993]. These small manageable information models are referred to as perspectives. Object oriented analysis techniques are used to build and integrate the perspectives into a single multi-faceted information model for a certain domain such as construction planning, conceptual design, spatial design, physical design, structural design, determine procurement systems, etc. Figure 5.5 shows a portion of the ICON construction object model.

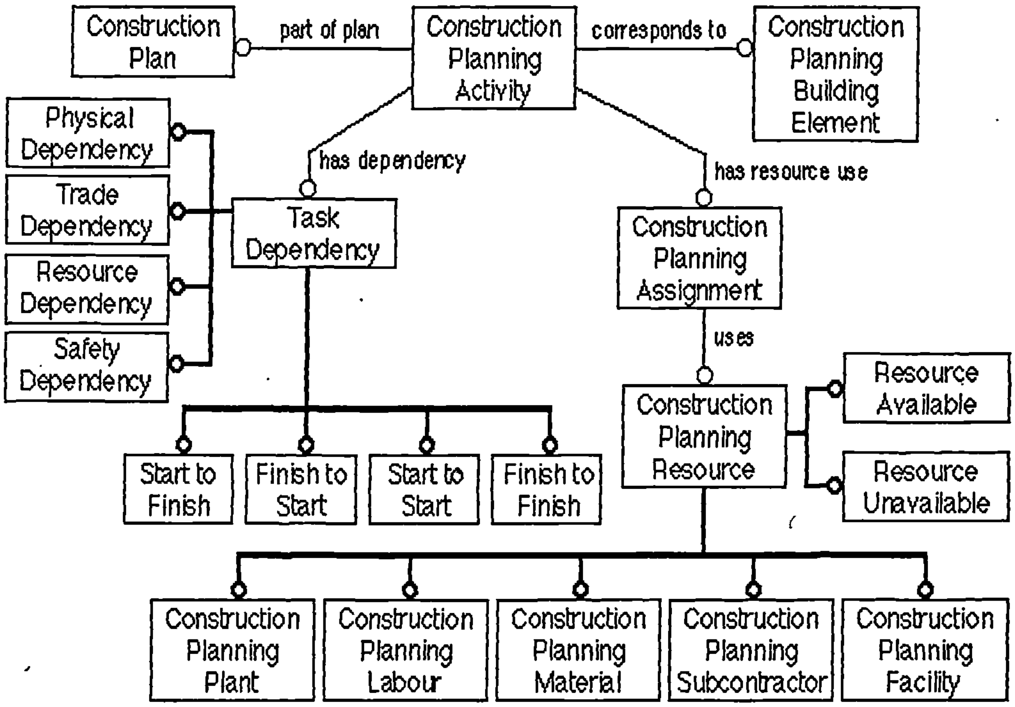


Figure 5.5: A portion of the ICON Construction Planning Object Model [Aouad *et al*, 1994]

The main significant advantage of the ICON model to the industry is its ability to view information from different perspectives as mentioned above. The inclusion of object types viewed from different perspectives but shared across different domains and abstraction levels is seen as a major step forward in integrating information throughout the construction industry [Aouad *et al*, 1994]. However, ICON model was developed through a top-down approach, which could lead to the implementation problems. The lack of perspectives of the lower detail domains such as attributes of a single beam creates data redundancy/duplication.

5.5.3 ATLAS

The ATLAS project aims at the development, demonstration, evaluation and dissemination of architectures, methodologies, and tools for Computer Integrated Large Scale Engineering (LSE) [Tolman *et al*, 1994]. The ATLAS models describe some of the relevant entities required to share and exchange meaningful data between different CAD systems used in LSE [Tolman & Poyet, 1995].

Like the RATAS model, the ATLAS model also distinguishes into abstraction levels, but in four levels only:

- *LSE Level* - contains LSE Project type Model (PtM), which is an IRMA-like sub model [IRMA, 1993] that describes which actor/agent performs which activity. Typical entities are resources, cost, time, etc.,
- *Sector Level* - contains Building Project type Model (BPtM), i.e. a specialisation of the LSE Project type Model. It describes building projects with all the entities commonly used in practice. It mainly consists of a set of Building System type Models (StM), each describing one system, e.g. Space System, Roof System, etc.,

- *Discipline Level* - or actor type oriented models, called View type Models (VtM) such as for architecture, structural and HVAC engineering. A VtM will show the entities and attributes that are relevant for the discipline in the different project life cycle stages, and

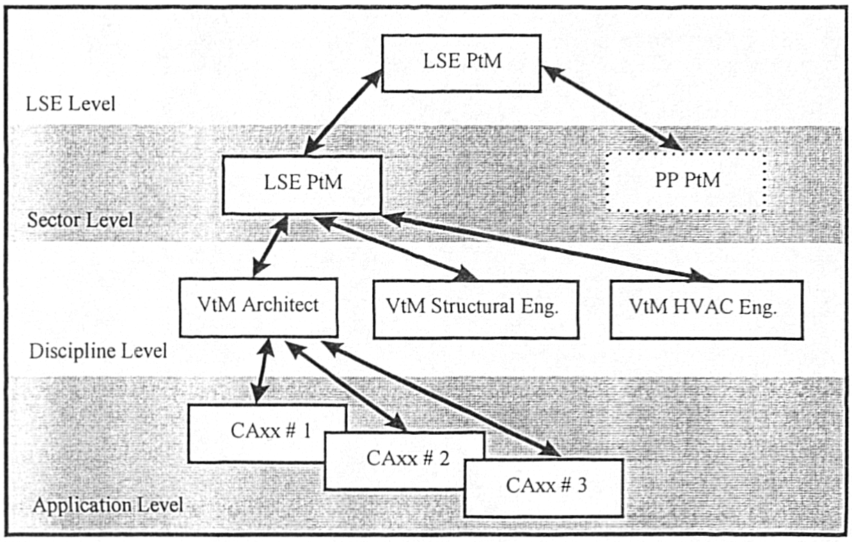


Figure 5.6: Relation between the different ATLAS models [Tolman & Poyet, 1995]

- *Application Level* - shows the individual CAD systems and their internal models. For example, the CAD systems used in Architecture (CAAD systems) communicate using one common data model, called an Actor View Model (VM). A VM contains a data model that describes the project/product model in the terms used by a particular discipline. A VM is an instantiation of a VtM.

The advantage of ATLAS model is that it was developed for the Large Scale Engineering industry. Although the LSE industry is very complex, some simplifications are required to manage the complexity. This is distinguished into three levels of communication: between actors of one discipline, e.g. between architects; between actors of different disciplines, e.g. between architects and structural engineers; between actors of different sectors of the LSE industry, e.g. between building construction and process plants. The model is intended to cater for different views, i.e. for use by different project participants such as architects, project managers, etc.

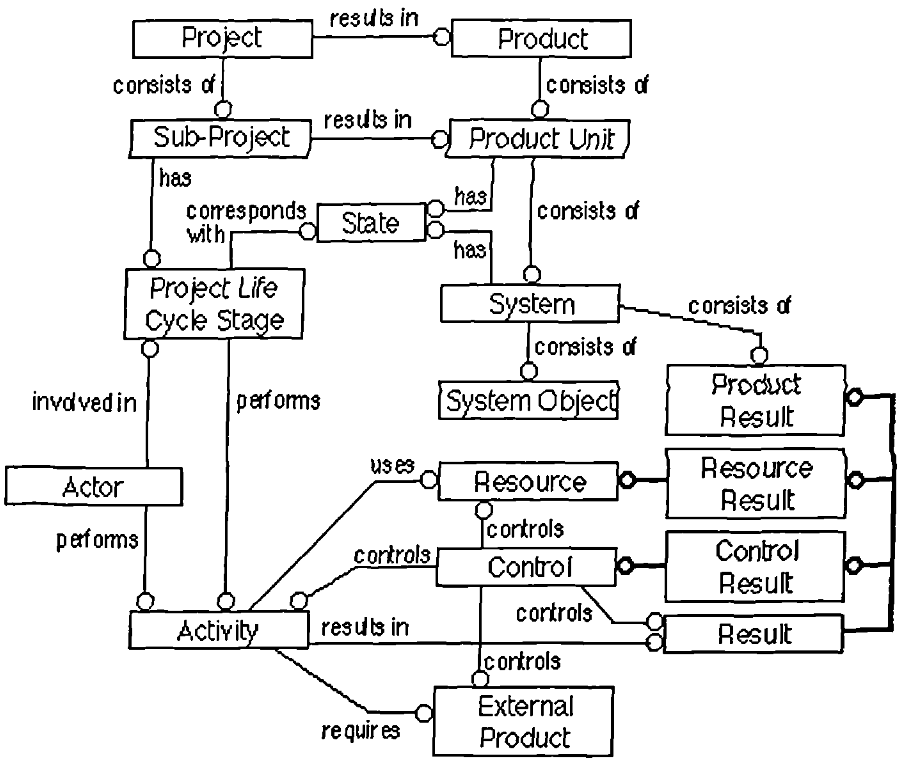


Figure 5.7: A portion of the ATLAS LSE project type model [Tolman *et al*, 1994]

Since ATLAS governs a wide area, currently the developed model covers only a part of the project life cycle. It contains a sub model that requires describing all the required project data about Actors, Activities, Controls, Resources, Time, Costs, etc. Figure 5.7 shows a portion of the ATLAS LSE project type model.

5.5.4 GenCOM

The General Construction Object Model (GenCOM), was developed as part of a PhD work carried out from 1989 to 1992 at Stanford University. Its aim is to improve the integration of project management software using standardised object-oriented models of construction projects [Froese, 1992]. The roles of standard models for the implementation of the integrated software packages, creating data exchange languages, and providing industry-wide information categorisation schemes were investigated. The results produced three models, each containing three elements:

Figure 5.8 shows some of the high level GenCOM object types. At the heart of the model is the representation of the project's physical components and activities that operate on the components. The activities represent the application of some action to some component using a particular method and a set of resources.

- *data model* - a formal plan for representing data in general,
- *domain model* or *schema* - description of how to represent the elements of the domain in terms of the data model. For example, domain model consists of a library of class definitions suitable for representing construction and project management information, and
- *project model* or *database* - using the domain and data models to express the actual data, e.g. relating to a specific construction project.

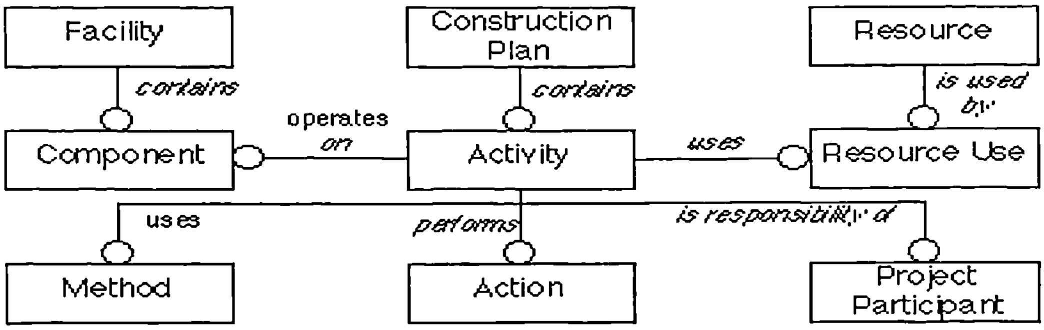


Figure 5.8: A portion of the high level object types from the GenCOM conceptual model of construction [Froese, 1992]

GenCOM is a model developed for a specific domain model, i.e. the construction and project management information. It does not govern all phases of the project life cycle i.e. from the inception to the demolition stages. However, all these phases share standard-domain model for constructed facilities. It was claimed that, this method would significantly improve the speed, quantity, and quality of the information where the project information could exchange [Froese & Paulson, 1994]. However, this method could create data duplications and data redundancies if all the domains were created in a separate domain model.

5.5.5 COMBINE

COMBINE (COmputer Models for the Building INdustry in Europe) is an international joint venture research project by eleven partners from seven countries in Europe. It aims to develop prototypes of the next generation of integrated building design systems (IBDS) based on available product data technology and cognisant of emerging ISO-STEP standardisation [Augenbroe, 1993; 1994]. The Integrated Data

Model (IDM) is the conceptual data model for the COMBINE project. The scope of the IDM can be globally described as building energy and installations information [Naderveen, 1995]. In COMBINE 1, the scope for the IDM is determined by the applications used in the project. However, in COMBINE 2, the scope of IDM is mainly determined by the Design Tools (DTs) since the validation of the IDM depends on the DTs.

Ten Design Tool Models have been developed separately. These are BCD (a building components database system), BRC (a building regulation checker), CETL1 (a costing estimation tool), DOCSET (a document management tool), DOE-2DX (a energy performance simulation tool), EN832 (a simple energy calculation tool), ESP-r (a complex thermal simulation tool), SUPERLINK (a lighting tool), TSBI (an energy performance simulation tool), and VENT (an air duct sizing tool). The conceptual data models for the DTs define the information requirements of the different DTs.

Although COMBINE is a big collaboration research project within the European countries, the COMBINE's Integrated Data Model (IDM) mainly focuses on building design tasks in energy/HVAC field. The models do not emphasise on the construction process and the project life cycle. Like the RATAS and ATLAS model, three levels are distinguished in which information modelling takes place in the COMBINE project; i.e. the design tool level, the integrated building model level, and IBDS level. Each level deals with specific types of information and employs different types of modelling formalisms, for example, the EXPRESS language.

5.5.6 OSCON

OSCON (Open Systems for CONstruction) is a research project undertaken at the University of Salford, UK, aiming to illustrate the benefits of using a centralised database as a means of integrating the information used by a number of participants within a construction project [Aouad *et al.*, 1998]. OSCON revolves around a central object oriented information model in which consists of two main parts. The first part is the domain models which support integration of information within a specific domain such as cost estimating. The second part is the core model which captures knowledge on how information is transformed from one domain model to another. OSCON demonstrated an approach to the integration of design and construction information using object orientation, integrated databases and VR.

5.5.7 WISPER

WISPER (Web IFC-based Shared Project EnviRonment) is a large research project undertaken at the University of Salford, UK, aiming of developing an integrated construction web site by adopting international standards in data models and communication protocols. The IFCs (mentioned earlier in Chapter 4) have been used as the core data model [Alshawi *et al.*, 1998]. WISPER also adopted distributed objects, CORBA (Common Object Request Broker Architecture) and the internet/intranet for delivering a truly distributed computer environment. The developed prototype has developed five applications: design, VR, estimating, planning and specifications.

5.6 Summary

The integration of the construction applications has encountered many problems. The main problem is data exchange/translation between applications. Traditionally, many interfaces have to be developed between all the application packages, i.e. one interface for each application.

Many of the data exchange problems can be overcame by having a central project model through which construction applications can access common or share information. This approach is known as product modelling approach. Through this approach, the number of interfaces required is significantly reduced to only one interface.

This chapter has discussed the principles of product modelling in an integrated environment. It highlighted the basic structure of the information modelling and the importance of product modelling in developing an integrated environment. Several product models which were developed by aiming towards Computer Integrated Construction (CIC) were reviewed.

From the reviewed models, it can be concluded that the development of a single product model should address the following:-

- using an object-oriented data model where object-oriented technologies are the most effective and powerful approach for representing large bodies of complex engineering information;
- emphasising upon the "abstraction level", i.e. the abstract of the building at the top level and the physical of the building at the bottom level;
- separating the domain into several object-oriented models, each describing the information needed to support a single well-defined activity;
- considering the implementation issues, i.e. to the details of the domain in the lowest possible level;
- considering the project life cycle in the project model starting from the inception to the demolition.

The next chapter will discuss the issue of the integrated environment whereby an integrated construction environment will be proposed.

Chapter 6

The Proposed Integrated Construction Environment

6.1 Introduction

The complexity and the vast amounts of information involved in any construction project, and the lack of standards have made the process of producing an integrated environment very difficult [Sanvido, 1992; Aouad & Price, 1993; Yamazaki, 1992; Faraj & Alshawi 1996]. On the other hand, the current systems used by industry are fragmented and can no longer serve the rapidly changing business demands. A study in the field of design and construction integration [Yamazaki, 1992], has identified the following as being the major problems in the current systems:-

- the system could not adjust to the business need when change is required.
- Basic information and knowledge about construction technologies is not shared between project participants.

- Interactive procedures where design or construction solutions can be evaluated at an early design stage have not yet been developed.
- No systematic evaluation and feedback of relevant data and information from construction sites.

An integrated environment can address such problems whereby all possible construction applications can be integrated under one environment. Such an environment should not be developed to suit a predefined set of applications, which could limit its usage. It should be designed to integrate "industry standard" software, relatively inexpensive to run, and flexible enough to accommodate the user's experience, in-house databases, as well as future needs and requirements. If this is developed and implemented effectively, it could assist in eliminating data redundancy, maintaining up-to-date information, improving accuracy and consistency of data transfer, and creating a suitable environment for concurrent engineering. The latter can significantly improve design solutions as users can quickly alter an application's output and examine its impact on other applications.

The complexity of the construction applications and the vast amount of information involved in a project have hindered the development of such an environment. Moreover, the lack of a high level structure (a strategic vision) of such an environment, has led to the development of a number of small integrated applications in various fields of construction. To enable such an environment to be established, a developer's view needs to be created to clearly define its requirements and to present the high level interactions between the various activities involved in the integration process.

This chapter addresses the issue of integration and proposes a strategic, but generic, framework for establishing an Integrated Construction Environment (ICE). The first section defines the description of a typical product model of the proposed framework at various levels of abstractions whereby a full process analysis of the various activities required to successfully establishing such an environment is explained. The terminology within the context of the proposed framework is discussed followed by a conceptual representation of the framework. The information flow within the Integrated Construction Environment and the implementation of the concept of the proposed Integrated Construction are also discussed.

6.2 A generic framework for integrated environment

Structured Analysis and Design Technique, SADT [Marca & McGawn, 1988] has been used to represent the proposed strategic framework for the Integrated Construction Environment (ICE), where design and other downstream applications can be integrated. The description of this technique is discussed in detail in Chapter 5. The proposed approach is a representation of generic activities along with their relationships, which demonstrate how downstream applications can be integrated with the central core data models. The conceptual model is represented in three layers, starting from the Context

Diagram down to level 1 process decomposition. The following sections explain the proposed approach using the IDEF∅ notations.

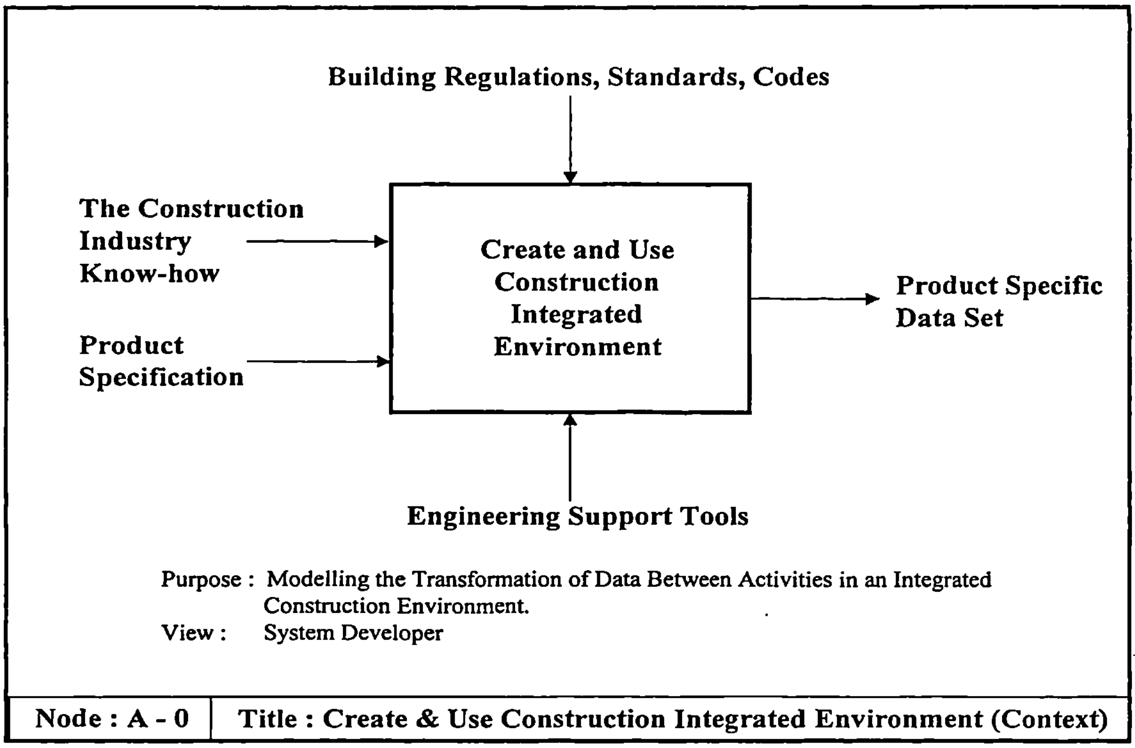


Figure 6.1: The Context Diagram of A-0

6.2.1 The Context Diagram (A - 0)

The IDEF∅ diagram of the overall Integrated Construction Environment (ICE) is shown in Figure 6.1. It represents the highest possible abstraction of the Input, Control, Output, and Mechanism (ICOM) which are required by and/or acting upon the ICE. The main input is the Construction Industry know-how (Design and Construction) and product's specifications. The former covers an abstraction of the construction industry know-how. The comprehensibility of this abstraction depends on the scope and the objectives set for the ICE. The latter, on the other hand, refers to a project specific information for which the ICE will be applied to, i.e. project specific data, which is normally obtained during the design stage of the project.

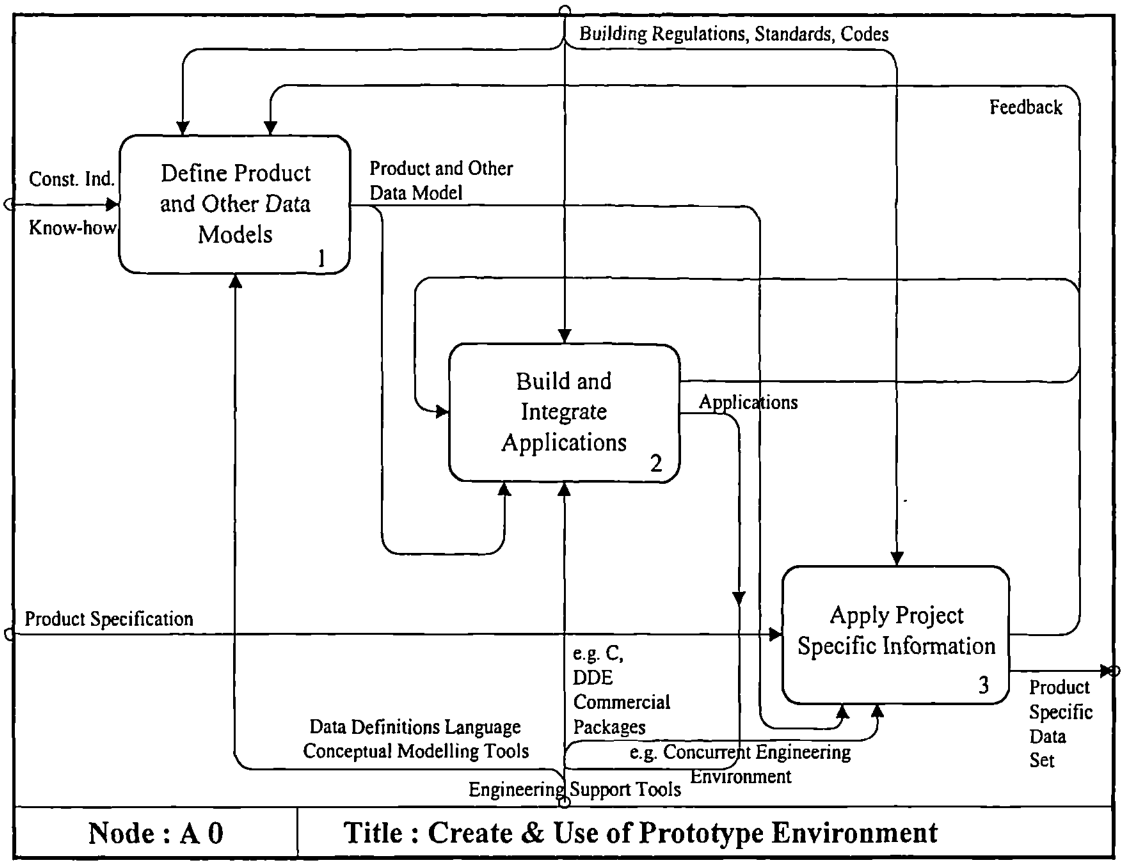


Figure 6.2: The decomposition of the Context Diagram (A-0) - A0

The execution of this process is constrained and controlled by building regulations, standards, procurement methods, etc. While the mechanism required for implementing the system is represented by the Information Engineering Support Tools such as: CASE tools, object oriented environments, data definition languages, etc. The output of the system is a product specific data set of which content depends on the main functionality of the ICE and its downstream applications.

6.2.2 The decomposition of the Context Diagram (A0)

The initial decomposition of the Context Diagram is shown in Figure 6.2. The first activity is Define Product and Other Data Models. This activity requires developers to identify the individual parts (models) which are needed to satisfy the scope and the objectives of the ICE's main functionality. The output of such an activity is a set of data models representing a product data model along with other data models required by the ICE such as construction planning model, estimating model, material's specification model, etc. The performance of this activity, i.e. the behaviour of the various models, is subjected to constraints such as regulations, standards, procurement methods, etc. Moreover, specific constraints are also applied when needed by the requirements of a specific project. This is shown by the Feedback arrow which is the output of activity 3, Applied Project Specific Knowledge. Activity 1, can be implemented using any information engineering support tools.

The second activity is the Build and Integrate Applications. This activity develops downstream applications and integrates them within the ICE. Downstream applications refer to those developed externally using commercial software packages and/or internally using the ICE information engineering support tools. The integration process requires a well defined structure for the central core data models so as information required by various applications can be easily accessed while altered or updated information can be easily written back to the product model. For example, to obtain information about a number of design elements (instances) from the product model, an application needs to know where this information is stored and the type of data structure, e.g. list, array, record, etc. Hence, the data models become part of the mechanism for this activity i.e. applications utilising the data models to access the relevant information. The main input for this activity is the product specific data sets which are generated by activity 3, Apply Project Specific Information. For example, in order to produce a cost estimate for a project, the estimating application requires specific information about that project which must be presented in the required format. The output of this activity reflects the output of the applications, which in turns can be used as a feedback to improve their future performance. Finally, the full implementation of such applications is subject to the relevant regulations and standards.

The final activity is Apply Project Specific Information. This activity populates and uses the previously developed data models with project specific information, i.e. instantiating instances normally from the design information. This implies that this activity uses data models as a mechanism to fulfil its objectives. It generates product specific data sets, as an output, and feeds them into activity 2 so that applications can be implemented. Therefore, once the project is defined, users can run the system to obtain the required output e.g. construction plans, estimates, Virtual Reality models, etc. Moreover, the output of this activity, i.e. project specific requirements, can be applied as constraints to the utilisation of the data models i.e. specific projects required specific data behaviour. The implementation of this activity, however, can only be carried out within an integrated environment, ICE.

In order to show the details of the above activities, each activity has been decomposed into its constituent's sub-activities and is represented in low level IDEF∅ diagrams.

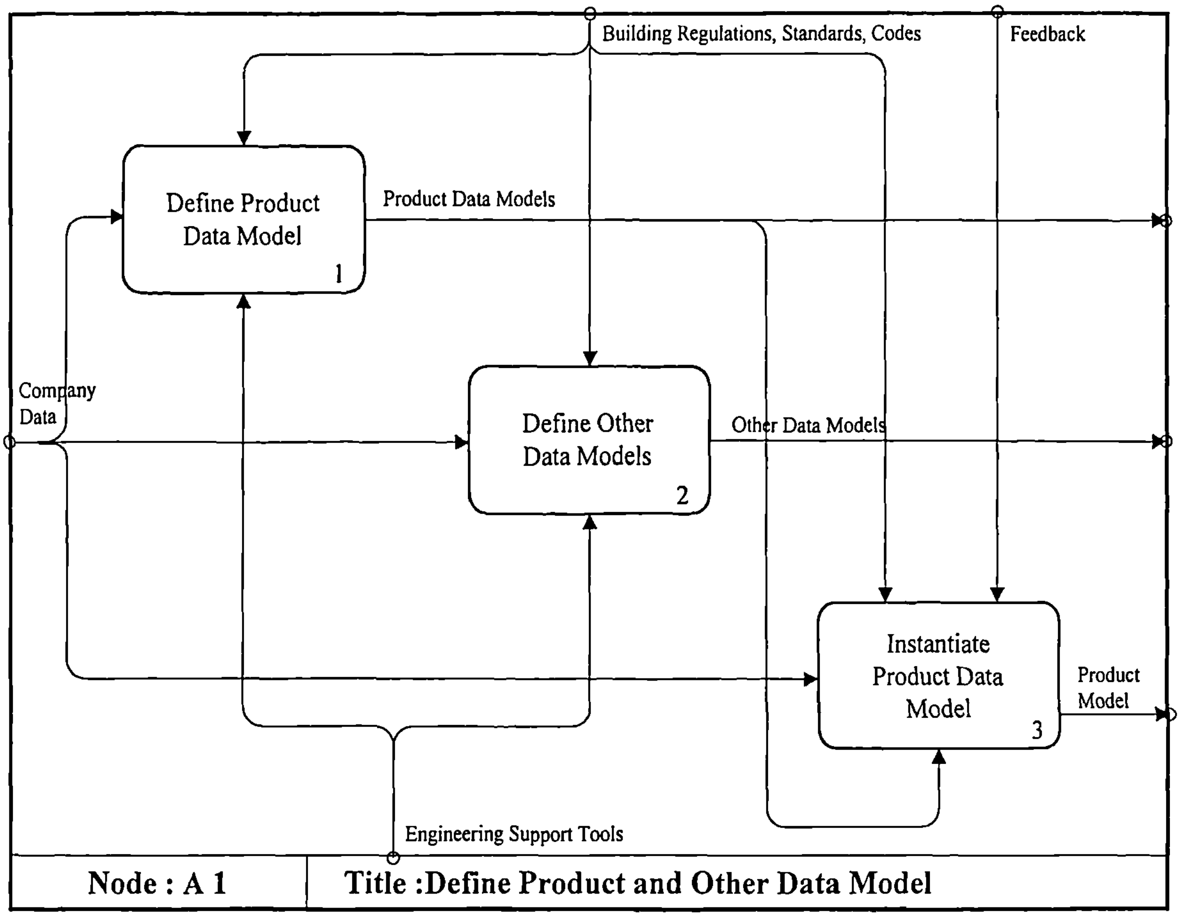


Figure 6.3: Define product and other data models - A1

6.2.3 Define product and other data models (A1)

Figure 6.3 shows the decomposition of Define Product and Other Data Models activity. Activity 1, in this diagram, is Define Product Data Model. It creates a product data model, based on the construction know-how, which represents various levels of data abstractions of the concerned building type, e.g. concrete buildings. Its main output is a defined and implemented product data model, which satisfies the scope and the objectives of the ICE. Its implementation is materialised through information engineering support tools. Finally, its behaviour is subjected to constraints imposed by the various regulations and standards.

Activity 2 is Define Other Data Models. This activity produces data models, which are related to one or more building types and fulfil the requirement of specific applications. The existence of such models does not depend on the existence of the product data model. For example, a Construction Planning Data Model, which contains information about construction methods, contractor's resources, planning knowledge, etc., can exist before a product data model is defined. It can also be used to plan other type of buildings, which are outside the scope of it's ICE. These models use the construction know-how as their main input and produces specialised data models, e.g. Cost Estimating Data Model, Site Layout Data Model, Construction Planning Data Model, Specification Data Model, etc. Such models are also constrained by regulations and standards, and are implemented using information engineering support tools.

The data models are simply static representation of data. However, the interactions between the various data models are controlled and carried out by the applications themselves. For example, a cost estimating application can interrogate the product model for instances of a building, which in turns can activate certain methods to generate their associated specifications from a specification model.

The last activity in this diagram is activity 3, Instantiate Product Data Model. This activity defines the status of objects by filling in the range of values required by their attributes. This information can be accessed either from on-line or in-house databases. When the data is entered into the product data model, i.e. utilising product and other data models as a mechanism, it becomes a product model, i.e. producing product model as an output. The performance of this activity is subjected to regulations and standards as well as the feedback coming from applying specific knowledge into the product model, i.e. activity 3 A0.

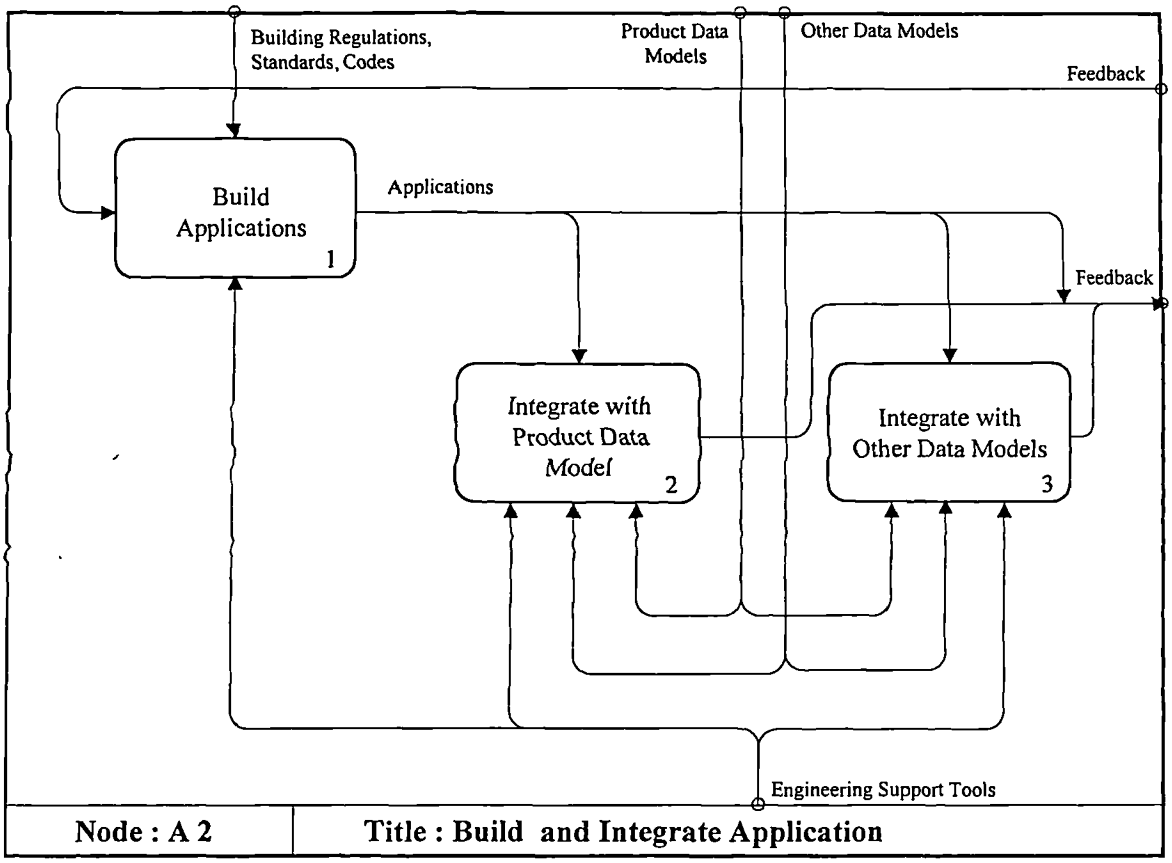


Figure 6.4: Build and integrate applications - A2

6.2.4 Build and integrate applications (A2)

This activity has been decomposed into three sub-activities as shown in Figure 6.4. Activity 1 is Build Applications. This activity refers to the process of developing any application that needs to be integrated into the ICE, i.e. with the central core data models. Applications can either be the existing ones or specially developed for such integration. In either case, they are implemented using programming languages, commercial software packages, DDE, etc. The development of such applications are subjected to further feedback arising from the use of the ICE which can be used as an input to modify/extend the applications. Finally any function performed by an application should comply with regulations and standards.

Activities 2 and 3 integrate the developed applications with the product data model and other data models, respectively. They are carried out through the development of interfaces between the applications and the environment with which the ICE is implemented. Thus, the data models and programming facilities act as a mechanism to perform these activities. Moreover, the applications themselves create constraints on these two activities. This is because the data defined by the data models must satisfy all the requirements of these applications. The outcome of these activities are used as future feedback mechanism for activity 3 A0, Apply Project Specific Information. Also, it is use as a feedback, which can either impose new constraints onto the data models, i.e. activity 1 A0, or be used for future developments of each application.

6.2.5 Apply project specific information (A3)

This activity is shown in Figure 6.5 where it has been decomposed into three activities. Activity 1 is Define Project Data. The activity's main function is to populate the product model with project specific information, i.e. transforming product specifications into Product Specific Data Sets. The latter sets are then fed into the various applications when required. Normally, this activity takes its input from a design application and in the format of building elements and their associated geometric information.

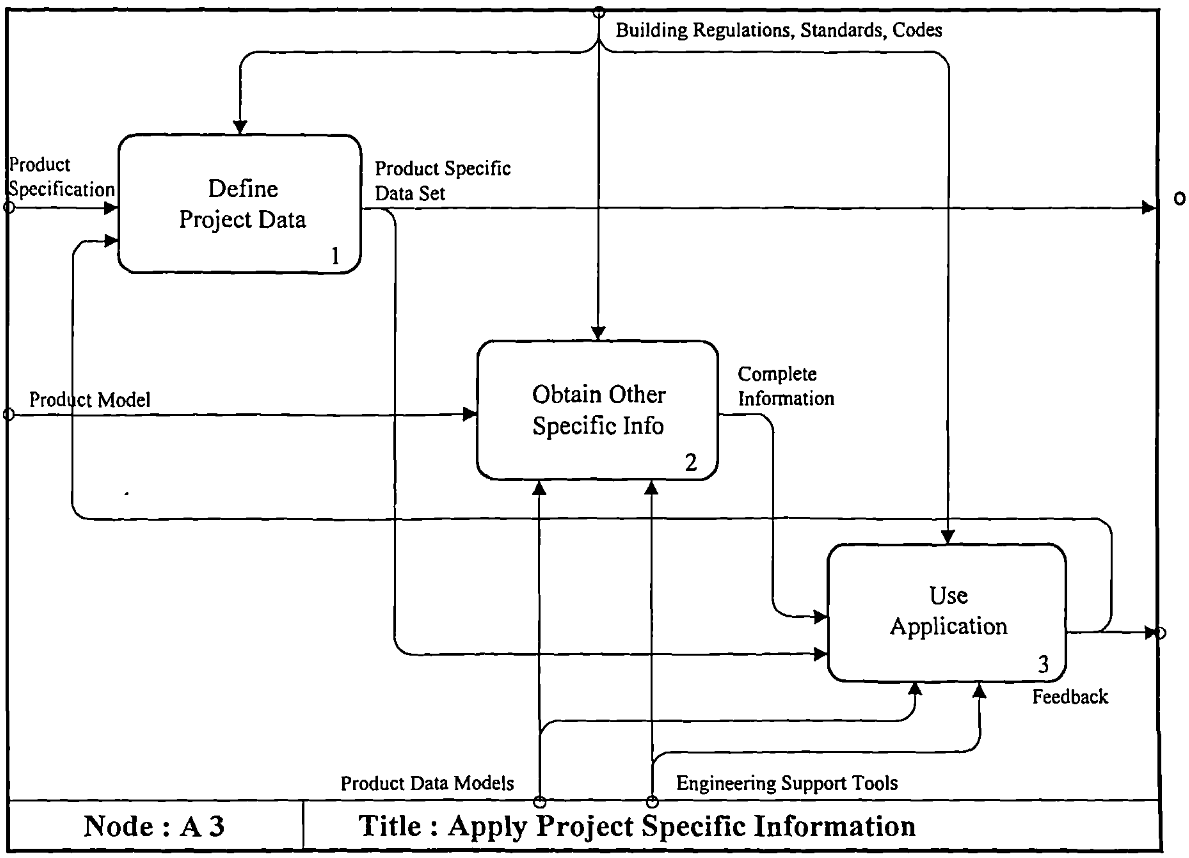


Figure 6.5: Apply project specific information - A3

Activity 2 is Define Other Specific Information. This activity obtains any other information, which is required by the downstream applications from the other data, models, i.e. where an application's data model does not fully provide the information required by the application. Therefore, the product model is used as the activity's main input. For example, for an application to determine whether a plant is required for a certain construction activity, it needs to know its size, location, site constraints, etc. before it can produce a recommendation. Interrogating the product model in the central core usually generates such information. This activity is normally an application oriented one where its output satisfies the applications' requirements. It is again controlled by regulations and standards and implemented using the information engineering support tools.

The last activity is activity 3, Use Application. It takes the previously generated complete information sets as well as data from the populated data models as an input in order to, allow the applications to perform their main functions. This activity is controlled by regulation and standards and implemented through either the information engineering tools or the applications' own environment. Its output is used as a feedback to enable users to change the Product Specification to improve the generated result.

6.3 Data structure

At the heart of the ICE, a number of data models may exist which support the applications to perform their functions during the project life cycle. Different ICEs may have different structures of these data models, level of abstractions, different coverage of the project life cycle, etc. Figure 6.6 shows an Express-G representation of typical data models covering the life cycle of a building. Three main parts are involved in this product data model; Specifications, Realisation, and Physical. These parts produce a complete definition of a class (object) through its life cycle. The Specification part contains data referring to the specification of the element (a project must have one specification). The Realisation part covers all the necessary data required to construct it (i.e. how the specifications are to be realised). An element can be realised in different ways e.g. many contractor can satisfy the specification in different ways. The Physical part holds geometric and other data that describe the physical status of the element.

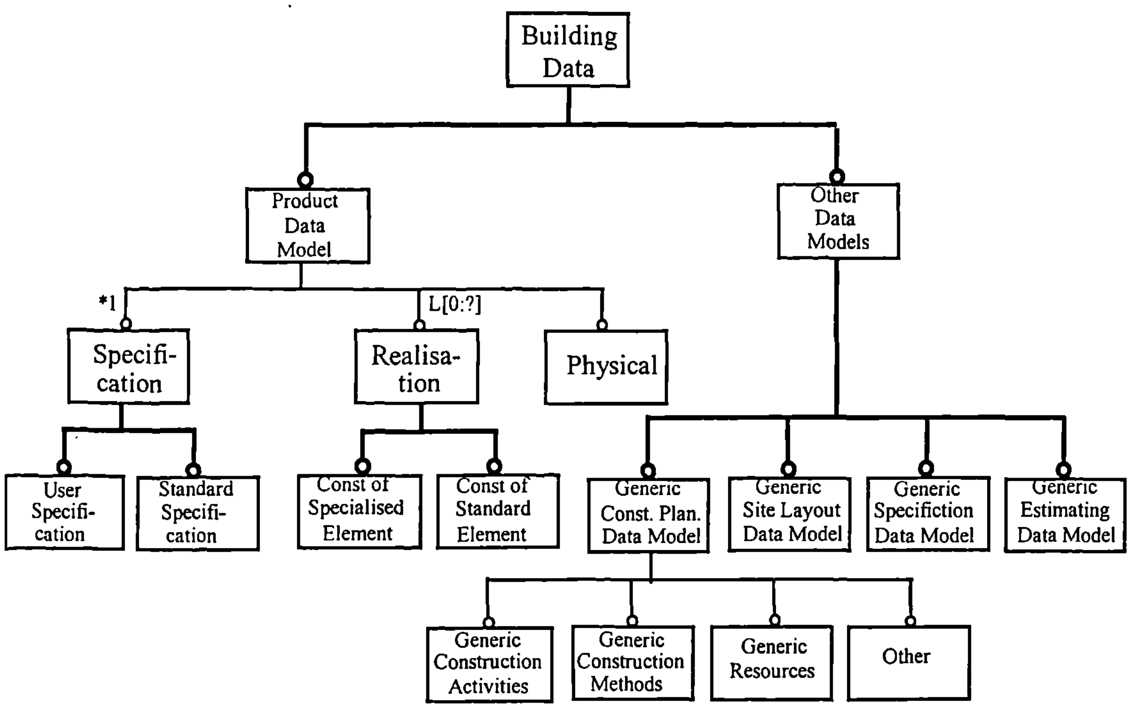


Figure 6.6: Typical models of the ICE

Other data models represent data required by other applications within the ICE such as Construction Planning Data Model, Site Planning Data Model, Specification Data Model, Estimating Data Model, etc. Each data model should be developed to achieve a well-defined set of objectives and to fulfil the need of a particular application. For example, a construction planning application requires a data model to support the information required by the application e.g. generic construction activities, resources available, construction methods, etc.

Information between the product data model and other data models may overlap and therefore the concept of data sharing must be applied. For example, Figure 6.6 shows that part of the specification defined for a particular element can be similar to that of the Specification Model. In such a case, similar data should be shared between the various data models using the data sharing principles. This is an important issue, which has to be addressed when designing and implementing such models.

6.4 The Implementation of the proposed framework

The above framework has been proposed for the Integrated Construction Environment (ICE) with the aim of co-ordinating the integration process between the various construction applications. The implementation of this framework has led to the development of a modularised central core whereby each application has its own data model.

The proposed structure of the ICE is shown in Figure 6.7. It consists of three main parts, i.e. the project model, software packages including interfaces with the project model, and external databases. The project model which is the modularised central core, is a model that combines both product and process views [Froese, 1995a]. Each module within the project model supports a particular application of the project life cycle and is supported by methods/events which are necessary to describe the module's behaviour and relationships with each other and with the external world, i.e. application software packages and external databases. Moreover, each module is underpinned by a knowledge base, which adds intelligence to its behaviour.

6.4.1 The Project Model

The project model comprises the building element data module and other application data and process modules. The building elements data module mainly describes the building's components and their behaviours. The extent and structure of this module depend on the scope, the context, and the main objectives of the ICE, e.g. an environment for concrete framed buildings may have different building elements data structure of that of steel. Other application data modules, on the other hand, represent data required by other stages of the building life cycle such as specifications, estimating, construction planning, site layout planning, etc. Each of these modules must be developed to fulfil the need of a particular construction application. For example, a construction planning application requires an application data module to support the information required by the planning process, e.g. generic construction activities, resources available, construction methods, etc.

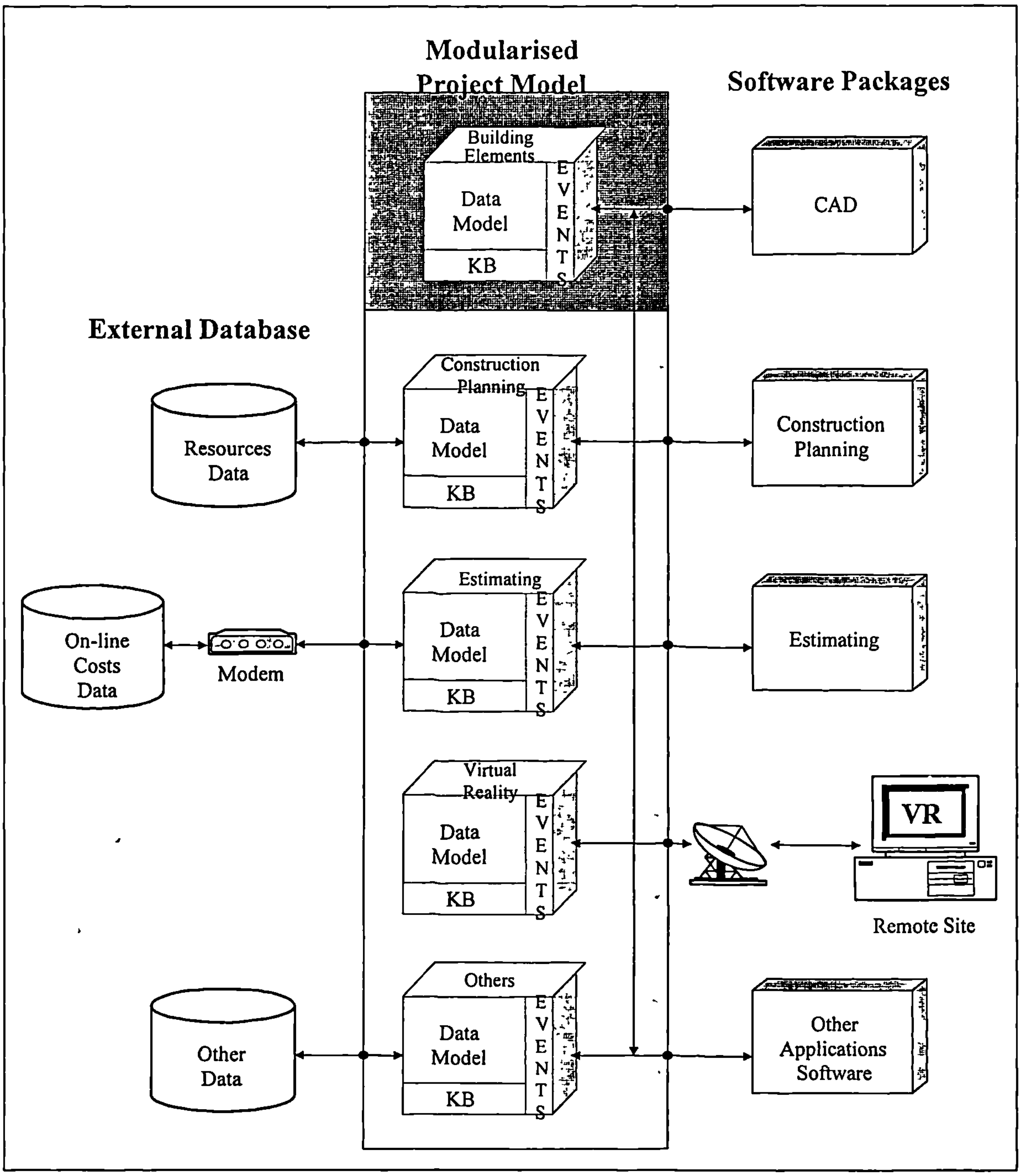


Figure 6.7: Conceptual representation of the ICE

The modules should be designed to complement each other and to maintain and share data in the most efficient way. In this approach, data related to a particular stage of the project life cycle is maintained separately from other data, but makes use of other data modules as and when required. For example, construction planning data module contains generic information about construction activities, methods, resources, etc. When it is activated, it refers to the building data module, where building elements of the current project is stored, to generate the project's specific construction activities.

6.4.2 Software Packages and External Databases

The second part of the ICE represents the construction applications packages such as CAD, construction planning, estimating, virtual reality, etc. Such applications software packages can either be external, i.e. stand alone applications packages, or internal, i.e. developed within the environment of the ICE. In either case, each application has its own user interface to manipulate the information, and a specially developed two way communication channel to transfer information between the application and its related application data module at real time. In Figure 6.7, dark circles on the communication line represent the interface. These application packages are completely independent from the project model.

The third part of ICE environment is the external databases. The project model can retrieve external information from external databases as and when required by the various involved modules. This process can be carried out directly by the modules or shared by a number of applications modules, e.g. estimating and construction planning applications may need to share cost data which can be retrieved by any of these applications, say from on-line databases.

6.5 SPACE

The above concept of the ICE was implemented by the AIC (Automation and Integration in Construction) research group at the University of Salford. A PC based integrated environment prototype, SPACE (Simultaneous Prototyping in An integrated Construction Environment) has been developed which aims to integrate the "industry standard" software through one object-oriented knowledge-based system where project information is shared and transferred automatically and transparently as and when required by the software/construction packages.

SPACE takes an alternative view to the traditional design and construction practices. It considers the project life cycle stages as applications, which can be, accessed simultaneously i.e. construction applications can be triggered off at any time. This provides all users i.e. clients, design team, contractors, planners, etc. with the opportunity to access the relevant project information at any stage. For example, clients can visualise the project in 3D and/or can examine whether the design solution meets their requirements and/or budget, while designers can investigate the impact of their design on construction thus giving them the opportunity to improve on the constructability of their design. In such an environment, the cost of carrying out "what if" scenario is almost null.

SPACE provides users with a multi-disciplinary computer environment where project information can be shared between the various construction professionals. Project information is stored in the project model where all construction software applications such as design, estimating, construction planning, bill of quantities, virtual reality, etc. are attached to. Information is transferred to the software packages as and when required. For example, design information, as generated by a CAD package, is transferred to the project model and can be accessed by all the attached software applications. Moreover, each of the individual packages can access standard data from external databases, as shown in Figure 6.7. For example, contractor's resources or prices required for estimating can be accessed from in-house and/or on-line databases.

At the current stage of development, SPACE integrates six construction applications which is divided into six applications modules as follows:-

- CAPE (Construction Application Protocol for data transfEr)
- SPECIFICATION
- CONPLAN (intelligent CONstruction PLANning generator for design rationalisation)
- EVALUATOR (project Estimate and interim VALUations (monthly) generATiOn in an integRated environment)
- INTESITE (INTElligent SITE layout planning)
- CONVERT (CONstruction Virtual EnviRonmenT)

6.5.1 CAPE

CAPE is a design application, which includes two main parts (detail of which will be discussed in Chapter 8 and 9). The first part represents a building element data module and the second part is the Object Interpreter Engine (OIE). The later is designed to transfer CAD primitives drawings, i.e. lines, points, arcs, into objects and populate the objects with the necessary information. It also provides an external data exchange file (DXF format) for each single element created to enable objects to represent themselves in other graphical environment such as CAD packages or VR.

CAPE is designed to serve all the application modules in the project model. It provides all the necessary design information such as co-ordination, dimension, etc. which is required by the downstream applications. For example, the construction-planning module uses the co-ordination and dimension of a beam to define the construction activities and calculate the duration for constructing the beam. The estimating module on the other hand uses the volume calculated from the dimensions of the beam to calculate the quantities of the concrete required by the bills of quantity (BQ).

6.5.2 SPECIFICATION

SPECIFICATION [Underwood & Alshawi, 1996] module generates the specification of the design element as soon as it is declared in the *CAPE* module. It produces the specification of each of the design elements, which are retrieved from standard materials/components databases. These databases have been developed based on WESSEX's cost database [Wessex, 1991]. The specification describes the design element components such as brick or concrete block, concrete reference, concrete grade, admixtures, reinforcement reference, etc. Each specification is only created once in the *SPECIFICATION's* module, which is later referred to by other similar design elements.

6.5.3 CONPLAN

CONPLAN [Hassan, 1997] aims to dynamically generate project specific construction plans, based on the required construction activities and the availability of resources, and to analyse the generated construction activities with the aim of evaluating the buildability of the design.

CONPLAN is an integral part of the SPACE environment. Its independent module is supported by knowledge in the form of method and rules, which enables it to respond intelligently to user requirements. The application uses design and estimating data from *CAPE* and *EVALUATOR* module to generate construction plans in three levels of details, i.e. detailed, executive and master plan.

6.5.4 EVALUATOR

EVALUATOR's [Underwood & Alshawi, 1996] aim is split into two, i.e. to generate the project estimates in the form of elemental BQ and to automatically generate interim valuation for monthly payments certificates from the construction plan. *EVALUATOR* generates the project estimates via its integration with the *CAPE* module and the *CONPLAN* module.

The basic estimating data and the monthly interim evaluation of the project are generated through the project model by utilising the information from the *EVALUATOR* module. *EVALUATOR* utilises virtual reality (VR) as an interfacing tool to simulate the project based on the valuation period.

6.5.5 INTESITE

INTESITE [Sulaiman, 1997] aims to produce the site specific layout information, i.e. the arrangement of temporary facilities for the selected resources; as derived by the CONPLAN and CAPE.

INTESITE generates site layout through its integration with *CONPLAN*, *CAPE* and *EVALUATOR* modules. Information regarding duration, dates, materials, etc. are extracted from *CONPLAN* module while information related to the design elements is retrieved from *CAPE* module. This module creates the site geometry, i.e. boundaries, access roads and gates using AutoCAD™ and transfers them into the project model.

6.5.6 CONVERT

CONVERT's [Alshawi & Faraj, 1995] aim is to support the applications which perform functions within the project life cycle by mapping the views of these applications to the virtual environment. *CONVERT* generates virtual reality models for the design elements created by AutoCAD/AEC™ at real time. It also enables the virtual objects to be interrogated.

6.6 The working environment

SPACE integrated environment is normally triggered off by feeding in project specific information through a design package, e.g. a CAD. As the design progresses, the design information is dynamically transferred to the building elements data module [Alshawi, 1994]. Once the data module is populated with the project specific information, users can run any other application package at any stage of the design. For example, the cost of the so far developed design can be dynamically determined by running the estimating application package. The estimating data module, in this instance, transfers the design elements along with their specifications and quantities to the estimating software package in order to produce either the total cost or cost break down of the current design. The generated costs can be altered at the estimating software package, if required. This alteration is then transferred back to the estimating data module where actions are triggered-off if the altered information is not feasible or does not comply with regulations, standards, in-house databases, etc.

Users can switch between various applications at any stage at any time. Software applications packages respond to requests from the project model with the output of these software packages, being limited to the amount of information supplied by the project model.

6.7 Benefits of SPACE

The implementation of SPACE has several benefits. It is seen to be the tool that would overcome current problems. Such benefits are:-

- *Data Sharing* – sharing of information between different construction applications. Thus enabling the integration between all the project life cycle stages.
- *Integrated Project Database* - the development of several modules using modularised approach has proved to be an effective and efficient way for integrating different construction applications. Applications can be added as and when required without effecting the overall environment.

- *Maintenance* - the project model can be easily maintained or updated. Such processes can be carried out on the individual application modules as and when required by the applications without interrupting the other applications.
- *Visualisation* - the use of virtual reality as a front end for all the applications within the integrated environment provides new dimension to the construction applications. Project information can be easily assessed and/or evaluated through this graphical interface.

6.8 Summary

The complexity of construction applications and the vast amount of information involved in a project have hindered the development of an integrated environment for construction. The lack of a high level structure (a strategic vision) and a full understanding of such environment has led to the development of a number of small integrated applications in various fields of construction. This chapter has addressed this issue and proposed a strategic, but generic, framework for an Integrated Construction Environment (ICE). This framework has been implemented in an integrated construction environment (SPACE). The structure of it is based on three main components; the project models, software packages, and the external databases.

The proposed ICE has successfully been implemented by the AIC research group at the University of Salford. Several applications, external and internal, have been developed and integrated with the central project model. Applications have been developed individually and independently from each other. The approach has proven to be essential to such development as it gives an excellent view on how the various parts of the ICE are integrated. Also, it significantly improves the understanding of the working environment, which could lead to meeting future user's needs and requirements.

The next chapter will discuss and highlight the issue and the development of data sharing in an integrated construction environment using object definition whereby the framework for an object's life cycle will be proposed.

Chapter 7

Object Definition in an Integrated Construction Environment

7.1 Introduction

Great efforts have been devoted towards the development of a flexible and comprehensive data structure for product models, with the aim of serving all possible downstream applications in CIC environments. However, product models cannot be populated with project specific information without clearly defining objects, which are anticipated to be transferred dynamically to the product model of the environment. The issues of generating information and defining their type and format which might be required by the downstream applications have not yet been fully addressed.

There are two main problems related to information sharing and information flow within and among the various data models in an integrated environment which involves object definition and data sharing. Object definition is the most efficient tool in which a number of attributes can be populated to an object in order to serve all possible construction applications efficiently and effectively. While shared data is a common data, which serves the interest of a number of objects in the integrated environment. These two issues are highly inter-dependent, i.e. the amount of data stored in each object can be significantly affected by the concept of data sharing. For example, if specifications of a "wall" object are shared between all "wall" objects, then it is not practically feasible to incorporate such data in all "wall" objects, i.e. such data must be shared between the "wall" objects.

Object definition and data sharing are therefore vital in delivering an effective and efficient integrated construction environment. Objects need to be populated with information as and when required, in a structured way, so that they can be accessed, maintained and updated quickly and easily. This chapter addresses the issue of object definition and data sharing and proposes a structured concept for object's life cycle, which aims of managing information and its flow within the integrated environment.

7.2 Problems with the implementation of data models

As discussed in the previous chapter, the STEP initiative is built upon the key concepts of Product Models (PMs) and Application Protocols (APs). Application protocols, if developed effectively, can play an important role in the development of future integrated environments. However, isolated development of such protocols in any implementation attempts could lead to the development of theoretical models which are difficult to implement as implementation problems can be of a completely different nature. The implementation of such large data models, including the proposed interaction with each other, can make the management of the development process almost impossible when implemented in an integrated environment. This is mainly due to the involvement of a large number of hardware/software/construction professional in the development and testing processes where the overall responsibility is shared between them.

Moreover, the absence of a strategic framework for the integrated environment within which the various data models interact may lead different developers/software vendors to interpret and implement the proposed application protocols differently. This could be attributed to the misinterpretation of the context for which the protocols have been developed. Nevertheless, dealing with a single application protocol, e.g. the design stage of steel framed buildings would not create great difficulties. However, the situation worsens when developers attempt to integrate or exchange data between various data models within one or more application protocols. This situation has been observed in a European project where all the proposed applications over the life cycle of a building were planned to be integrated through one product model, i.e. an integrated database. The demonstration has been focused on file transfer from one application to another [CIMSTEEL, 1997].

Data exchange within and among the various data models depends on the structure of the data models and context of the environment, which they are prepared for. Objects need to know the type of information and knowledge that they have to keep within their frames and those that have to be shared, i.e. common data. This can not be done without a clear understanding of how these objects are populated over their life cycle. It is very likely that objects inherit different types of data at different phases of their life cycle depending on the structure and context of their environment. It is therefore, important to establish a well-defined framework for the integrated environment before attempting to establish "standards" for object's definition and data sharing. This can significantly improve the implementation process and the development of data models.

At this end, it is important to develop an agreed structure for an integrated environment in parallel to the development of the data models. This can significantly improve the implementation process and can yield an important feedback on the developed application protocols.

7.3 Objects and dependent-objects

In an integrated environment such as the proposed ICE, which has been discussed in the previous chapter, the term "objects" is referred to the elements that define the context and the scope of the environment. Dependent-objects meanwhile, are those whose existence depends on the presence of the main objects, i.e. those objects which hold data relevant to a single object over multiple applications. In the ICE, for example, objects are representatives of the building elements such as walls, columns, slabs, etc. and are normally stored in the design data module. However, when these objects are used by another construction application data module, the latter may generate specific data sets, which are relevant to the main object. Such data may require the instantiation of new objects in the other application data module. Such objects are referred to as dependent-objects.

If the construction of a wall object requires a particular type of crane, a "crane" dependent-object may be created by the construction planning data module, within the integrated environment, so that it can calculate the productivity and duration for the wall's construction activities. Depending on the aim of the integrated environment, the "crane" dependent-object may be deleted when the wall object is constructed or possibly transferred to another integrated environment in order to maintain control over the contractor's resources. This indicates that objects are kept in the environment until the user deletes them, while dependent-objects are created or deleted by the main objects.

7.4 Object's definition

The definition of the object, i.e. the process of populating the object with data, begins and continues until the end of its life. This information must be of the type and format required to serve all possible construction applications effectively and efficiently, i.e. it must enable objects to respond to the needs of different users at different stages of the building realisation cycle. Due to the complexity and duplication of such information, it is not practically feasible to populate each object with what it requires over its life cycle. For instance, it is not practical to populate every "column" instance with its specification, especially if there is a large number of columns and they all share the same specifications. This will create data management problems, i.e. data control and maintenance, and data modelling difficulties.

Within the context of the proposed ICE, an object is defined as a building design component that constitutes an integral part of the project, e.g. wall, beam, floor, etc. An object definition is the description of the design element in terms of its descriptive and basic features, behaviour, properties, topological relationship, etc. Such a definition is a vital pre-request to any successful related application within the proposed ICE. Its existence will be recognised by the ICE and therefore by all other downstream applications. Thus, in order for such objects to respond correctly and efficiently to the needs of different users at any stage of the project's life cycle, each object has to contain the necessary information and knowledge to serve all possible downstream construction applications.

Object definitions must be comprehensive and efficient if objects are to behave adequately, efficiently, and intelligently over their life cycle. This definition is highly dependent on the aims and the structure/context of the environment within which the objects are serving, i.e. the object must be able to behave in a manner that meets the environment's objectives. For example, the type and format of the information stored in an object created in an estimating application will be different from that created in a graphical application.

Objects are normally inflated with data during their life cycle. At each stage of the life cycle, objects are supplemented with necessary data in order to enable them to address the need of all construction applications. Depending on the structure of the integrated environment and the applications concerned, objects are expected to have different sets of data. In this section, the attributes that objects inherits during their creation phase, i.e. in a CAD package which is considered to be the main input for project's information in the proposed ICE will be discussed. The attributes are limited to the architectural and structural elements only.

As objects are created in a CAD package and before they are transferred to the central core of the ICE, they need to extract data from the environment in which they were created. These data can be divided into two groups, i.e. global data and specific data. In Table 7.1, Global Data are attached to all elements while the specific data are attached to specific building elements. Each single data, which is attached to an object, serves specific needs in the ICE and is stored in the most efficient way.

7.4.1 Global data

This section explains the global data and their expected usage in the ICE.

a) Associated Element

The Associated Element attribute is the list of elements that is associated with the object. For example, a door and a window can be built in a wall. The wall object will have the associated element of the door and window whereas the door and window objects will have the associated element of the wall. This relationship can easily be captured during the design stage, i.e. in the CAD package, and can play a significant role in serving several downstream applications. For example, in an application such as taking off or costing, the net wall area can easily be calculated by referring to the associated element attribute.

Global Data	Specific Data	Building Elements
Associated Element	Cavity Thickness	Cavity Wall
Co-ordinates	Inner Leaf	
Level	Outer Leaf	
Rotation Angle	Structural Function	
X Dimension	Depth from Ground	Foundation
Y Dimension	Going Length	Staircase
Z Dimension	Landing Length	
	Riser Height	
	Pitch Angle	Roof
	Area	Slab

Table 7.1: Global and Specific data of object definition at the creation phase

(b) Co-ordinates

Co-ordinates refer to the x, y, and z positions of the starting point of the object. In the proposed ICE, the centre point of the beginning of the design element has been selected while the co-ordinates (reference point) of the whole building are based on the bottom left corner. This attribute will assist users in locating design elements with respect to a reference point. This is absolutely vital to applications such as material storage, allocating resources to construction activities, selecting construction methods, etc.

(c) Level

This attribute defines the level of the design elements, i.e. the storey number at which the element is located. This information can be of use to a number of applications in particular costing and planning.

(d) Rotation Angle

Rotation angle is measured counter clockwise from the positive direction of x-axis. It is a basic geometric data which is used either to find the orientation of the design elements or for mathematical calculation of distances. For example, in an application such as site layout planning, it is necessary to find the orientation of the building with respect to the land. Meanwhile in construction planning, the orientation of a building element can have an impact on the volume of work and the resources allocated to the construction activities.

(e) X, Y, Z Dimension

The x, y, and z dimension are the geometric data of the element, i.e. the length, width, height, depth, and thickness. For consistency, its x, y, and z dimensions describe each element. Figure 7.1 shows two elements, i.e. a wall and a slab. The wall object, for instance, has the x dimension for length, the y dimension for thickness, and the z dimension for height, whereas the slab has the x dimension for length, the y dimension for width, and the z dimension for thickness. This geometric data is vital for basic calculation such as areas and volumes.

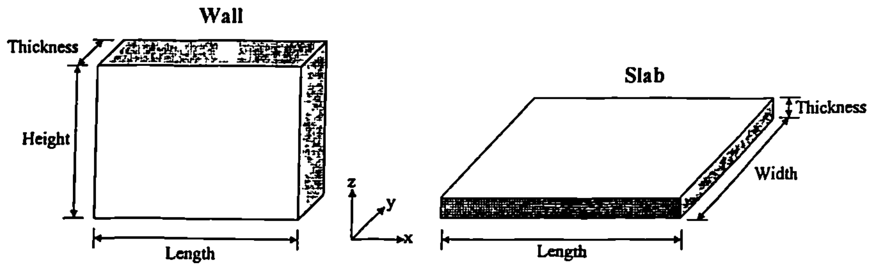


Figure 7.1: Typical building element geometric information in 3D

7.4.2 Specific data

This section describes the specific data, which are required by specific objects to enable them to serve all downstream applications in the ICE. This data is explained by considering these specific objects.

(a) Cavity Walls

A cavity wall has a cavity thickness, an inner leaf and an outer leaf as a specific geometric information. Its function can either be structural on non-structural, i.e. load bearing or non-load-bearing. The latter is normally defined at the design phase, i.e. at the CAD package. The cavity wall's geometric data are used to select the wall's specification, which in turn can influence a number of applications such as costing, planning, material type, etc.

(b) <u>Foundation</u>

In spite of the global data, foundation needs to store data relating to its depth below the ground for sub-structure works. This is an important attribute especially for a construction planning application where construction activities can depend on the depth of excavation.

(c) <u>Staircase, Roof and Slab</u>

These objects require specific geometric information due to their complex shapes. For example, the area of a slab can easily be extracted from a CAD package rather then calculated in a PM. While staircase requires landing length, going length and riser height to enable costing application, for example, to calculate the quantities of different types of materials required for constructing it.

7.5 The proposed framework for object's life cycle

In the proposed ICE, design information is transferred to the design data module as soon as it is created. Once drawing primitives of a design element are converted to a high level representative object, the life of this object starts with the creation of a representative instance in the design data module. This object is then accessed by the construction applications. Such applications may change the object's physical characteristics or status, may create new relevant data, or may delete the object from the environment. Such changes occur over a period of time and possibly independently from each other over a number of stages. The concept of the object life cycle has, therefore, been used to understand and formalise the behaviour of objects from creation to deletion.

This section introduces a new structured framework for the object life cycle. This framework has been developed with the aim of managing the process of populating objects with information within the ICE. It provides developers with a comprehensive and consistent methodology which is vital for co-ordinating the implementation process, future maintenance and updates. The proposed framework for the object life cycle is described in four phases: create and amend; supplement object with data; use object; and decommission object. Figure 7.2 is an IDEF∅ diagram, which illustrates these four phases in detail.

7.5.1 Create and amend object

The first phase is concerned with the creation of an object in an application. An object can be created in any construction application such as CAD packages, databases, planning packages, etc. When an object is created in an application, it is constrained by the user/firm's experience as well as building codes and standards. These constraints can highly influence an object as they control the type and the amount of information, which can be attached to the object during the early stages of its life. However, once an object is created in an application, basic information that is relevant to that application is attached to the object. For example, if a wall is created in a traditional CAD package, the geometric information is automatically attached to the wall object. Alternatively, if the same object, i.e. a wall, is created in an estimating package, it will have the information which is required to calculate its cost i.e. unit rate, quantities, resources, etc. In general, this phase populates objects with their first definition, which is referred to as the basic object definition.

7.5.2 Supplement object with data

The second phase of the life cycle of an object is to supplement object with data. In an integrated environment such as that proposed earlier, this phase is carried out in two stages. During the first stage, the object is associated with its relevant data module in the ICE and is then attached to its specialised class within that data module. This activity is constrained by the capability of the implementation package(s), i.e. if an object is transferred between two stand-alone applications then it will be constrained by the communication utilities provided by the packages selected. Thus, this stage can have a significant impact on when and how data is transferred and in what format. At the second stage, the object is populated with its parent's class information utilising the object oriented features of the ICE i.e. inheritance and polymorphism. The type of inherited information depends on the scope and the structure of the data module.

When objects are transferred to the project model of the ICE, they are normally attached to a design data module, which is an elemental representation of the type of construction project under consideration. The type and amount of information and knowledge inherited by the transferred object depend on the structure and knowledge built within that data module. Normally, this type of information is of a descriptive nature where an object inherits its identity, classification, and general behaviour.

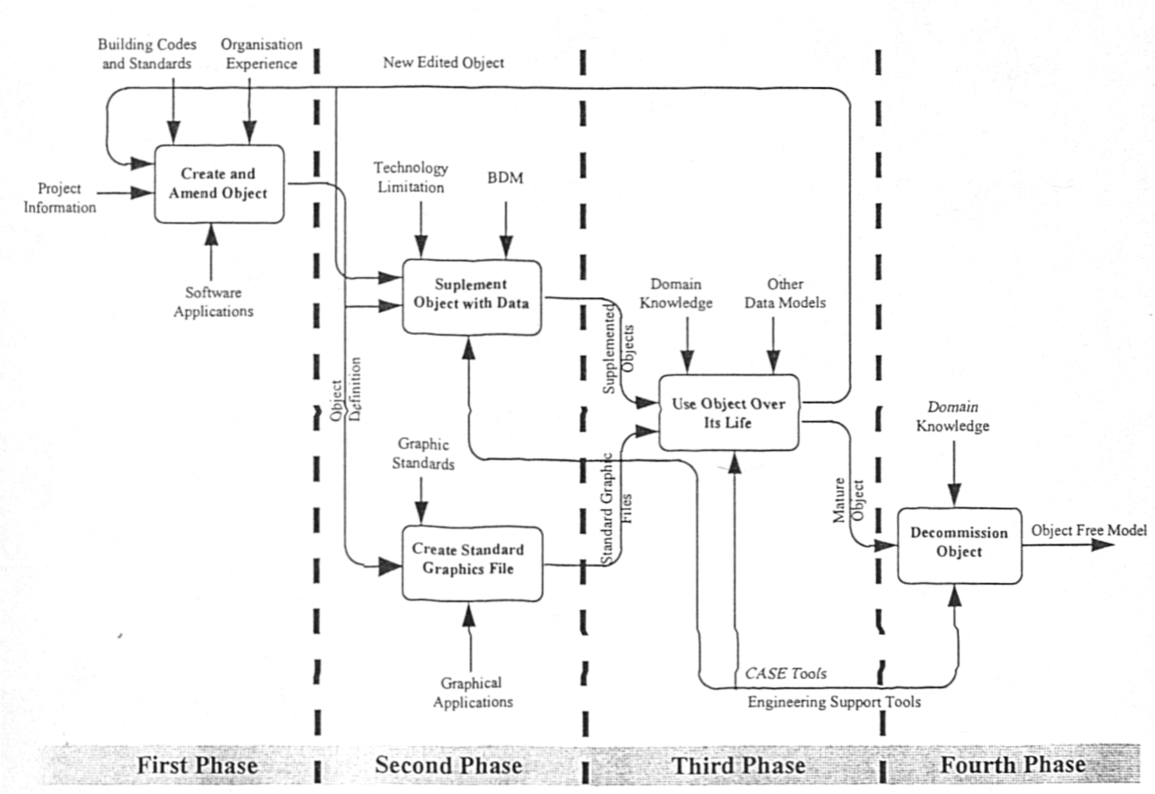


Figure 7.2: Process modelling of object's life cycle

If an application within the ICE, such as Virtual Reality, requires a graphical representation of an object, the newly created objects must inform the environment which created it to generate its standard graphics file. It must also retain a reference to that file. For example, if an object is created in a CAD system, the object can request this application to create a DXF or an IGES file and then store a reference for it. In some cases, objects may refer to a relevant standard graphics file, which may be part of a library of graphical objects. This process enables objects, at a later stage, to represent them graphically in any graphical package.

7.5.3 Use object

Once an object is supplemented with data after its creation, its existence will be recognised by other applications within the ICE. Only during this phase does the object become visible to the rest of the application in the integrated environment. This is mainly because the object needs to be instantiated in the project model first, before it can be accessed by other applications. At this stage, objects are expected to have all the necessary information to serve all construction applications within the ICE. However, the way in which objects communicate with each other is constrained by the domain knowledge, i.e. the context of the ICE and the structure and knowledge of the application data modules within the ICE. For example, a particular "space" object can only access space-planning knowledge if the space planning data module existed.

In general, objects at this stage are populated only with references, which refer them to other related data. This implies that objects are used either to:-

- Provide descriptive information including referencing to its own graphical (other) files;

- Generate relevant information within other application data modules. Objects normally instruct other application specific data modules to generate specific data, which are of relevance to the objects. Objects, in this case, only maintain a reference to that information;
- Share common data with other objects. Objects can refer application specific data modules to common data by supplying them with a reference.

During the third phase, the status of an object can be altered due to a request from an application. This can be carried out by altering one or more of its definition features. In this case, the object automatically adjusts itself by going back to either phase one, if the changes affect its basic definition, or to phase two, if the changes affect the supplemented data. In either case, the object will adapt to its new status. Such changes can be documented within the object to enable the object to return to its old status if needed.

7.5.4 Decommission object

The last phase of an object life cycle is the decommission phase. Users can terminate the life of an object by deleting it from its data module. This results in an end to the existence of the object and to all its relevant information, including those generated by the object e.g. its dependent-objects. The output of this phase will be an object free module.

7.6 Data required by an object over its life cycle

Figure 7.3 is an Express-G diagram representing the data required by an object over the four phases of the life cycle. An object can have four groups of information reflecting the four phases of the life cycle (matching the activity diagram, Figure 7.2). These are the created data, supplemented data, used object data, and decommissioned data. The first group of information, the created data, is data related to the basic definition of an object. The latter represents data generated at the time when the object is created in an application. This could include basic descriptive information, such as length, width, start co-ordinates, finish co-ordinates, area, orientation, etc., and specification data which is required to give the object an independent identity. The latter could include the type of object such as wall, beam, floor, etc., and other object specific information such as special structure type, special material, special thermal property, etc. The type and format of the basic definition information depend on the application package used and may vary.

The second group of information relating to an object is the supplemented data. This group of information represents data, which is inherited by the object during the second phase of its life cycle, normally from the design data module. Objects usually inherit their functional classification e.g. a wall can either be a load bearing or non-load bearing wall. They also inherit their general non-geometric data from the design data module such as type, behaviour, codes and standards. This allows objects, for example, to calculate their cost, and design/construct themselves, referencing other application data modules, etc. This process is completely dependent upon the method these data modules are structured within the ICE.

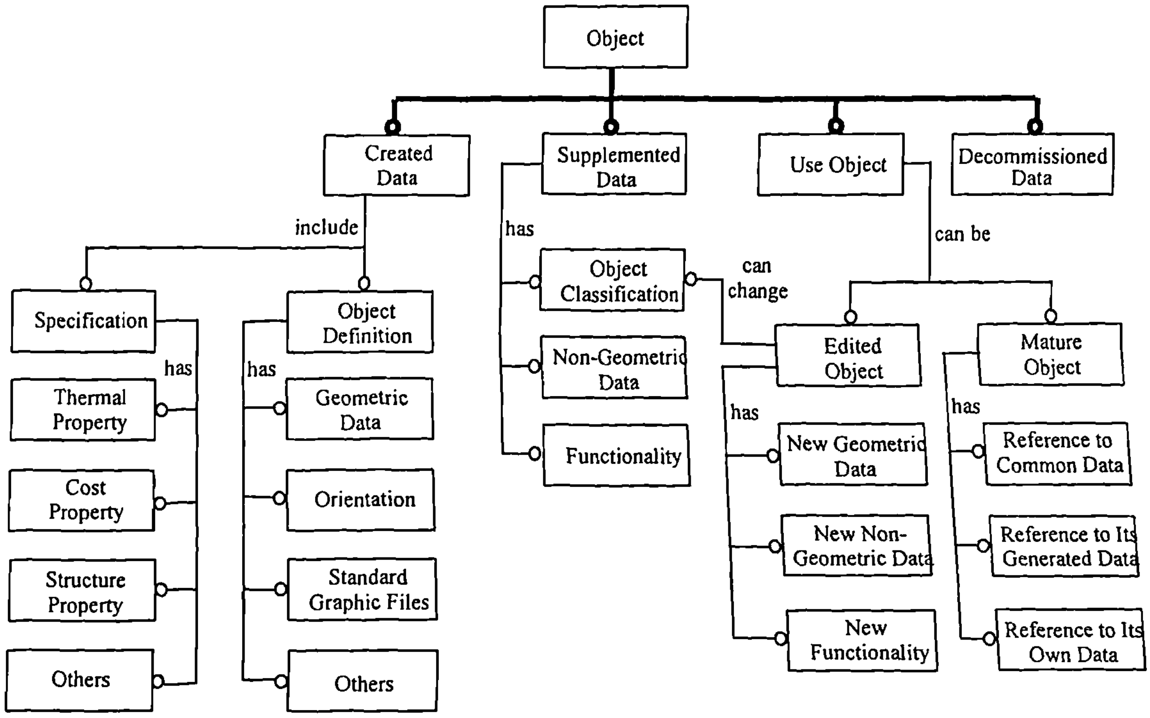


Figure 7.3: Data required by the object's over their life cycle

Once an object is supplemented with data, it should be capable of serving all construction applications. This is not to say that objects are loaded with all the information and knowledge required by other application packages, as this is impractical and very difficult to achieve. The previously discussed structure for the ICE, along with the proposed life cycle, enable objects to hold optimum data thus giving them a total flexibility for interrogating and integrating with other data modules.

The third group of information of an object represents data generated during the life of that object within the ICE. An object can serve other applications in three ways (as previously explained). First, it can use or refer to its own encapsulated data and knowledge. In this case, if an application interrogates a particular object requesting basic and/or descriptive information, the object responds by sending the required information or points to its own data i.e. standard graphic files.

Second, an object can generate a relevant set of data within another application data module and retain a reference for it. This usually occurs when an application data module requests information about a subsequent set of events, which can be caused by the object. For example, if construction planning information needs to be generated for an object, the construction planning application data module requests a list of construction milestones from the object in the design data module before generating the specific construction activities and their associated resources. The object concerned then sends the construction milestones data to the construction planning data module to generate the specific construction activities from its generic set of activities, establishing the relationship between them, allocate the necessary resources, etc. The main object retains a pointer to that part of the construction planning data module for future reference.

Third, an object can refer to the common data. This occurs when an application requests specific types of information which are not maintained by the object itself, but stored in a different application data module to allow for data sharing. For example, to estimate the cost of a wall, the estimating data module requests the specification of the wall concerned from its object. Because this type of information is normally stored in a specification data module, which is shared by all wall objects, the wall object points to its specification within the specification data module. As a result, the object retrieves the required data from the specification data module and passes it on to the estimating data module.

At this stage, the object may also be requested, by an application, to alter a few of its basic features and subsequently may change its existing status. For example, an application might request a wall object to change its basic materials from bricks to plaster board. This may have direct implications for other specifications of the wall and possibly on the geometric information, e.g. the height of the wall. In this case, the object becomes a new edited object and can inherit a different set of information but maintains its original functionality and type, i.e. the object is still functioning as a wall but with different properties. This illustrates that editing an object can only affect its features and not its main identity i.e. a wall object cannot be altered to a window object. Objects become matured as they are used by the construction applications. This maturity normally refers to adding more references to the object as described above.

The decommissioned data is the last group of information about an object. When the life of an object is terminated, it can either be deleted or suspended from the module. The former means that both the object and its related data no longer exist in the ICE, while the latter makes the object obsolete i.e. when an object physically reaches the end of its service life. In this case, the object is preserved in the ICE with all its data.

These four groups of information reflect the necessary information, which objects within the ICE, need in order to maintain their role in serving all construction applications. The amount and type of information populated into the objects at each stage depends on the scope and objectives of the ICE. However, this framework gives a clear methodology on how and when objects are populated and accessed throughout their life cycle.

7.7 Multiple view provider

The proposed framework for the object life cycle, within the modularised project model in the ICE, provides an excellent tool to satisfy the widely debated issue of providing multiple views. Once an object is instantiated and has entered the third phase of its life cycle, it can respond to different application data modules according to their needs and requirements. The application data modules, from the user's point of view, can illustrate the realisation cycle of a building type such as design, tendering, construction, refurbishment, etc. (as previously explained). Each of these stages refers to a particular application data module, which is a specialised domain of a particular profession (user). Therefore, each application data module can be considered as a view whereby a view is equivalent to an application data module. Views can be as complex and detailed as those corresponding application data modules. Figure 7.4, illustrates this principle diagrammatically.

To illustrate this concept, let us consider a "column" object. If this object existed in the design data module and in three application data modules such as structural design, construction planning, and estimating which support the integrated environment, then the object can provide three views which correspond to these application data modules. If a structural design package accesses the structural data module with the aim of designing the object "column" then the structural data module interrogates the object "column" for specific information and then displays or sends this information to the structural design package. On the other hand, if a construction planning package accesses the construction planning data module requesting specific construction activities which are required to construct the object "column", the "column" object responds to this application data module by sending its milestone construction activities. The construction planning data module then processes this information and displays or sends them to the planning package.

However, for some views it may not be possible to extract all the information required from one single application data module. For example, if a cost estimate is required for the object "column", the cost will depend on the construction planning data module as well as the specification data module. Thus, to provide the cost view to users, the estimating data module requests specification from the object "column". The object "column" points to the specification data module where its specification is extracted and sent to the estimating module. This information is then shared between the estimating data module and the construction planning data module. The latter uses this information to decide on the best construction method required to construct the column and then allocates the required resources. All this information is then given to the estimating data module to provide the required cost estimate.

This process requires a structured procedure in order to carry it out effectively and efficiently. "View Moderators" have been introduced for this purpose. These moderators are defined as a dynamic collection of methods, which is required to satisfy a particular view within an application data module. However, their existence and complexity depend on the status of the environment when the view is requested. In the above example, if the specification of the object "column" has not yet been generated, the cost estimate view moderator asks the object to generate its specification. If the specification has already been generated then the view moderator will directly access such information. View moderators are normally stored in the application data modules concerned and are triggered off by the application packages.

7.8 Definition of a wall object

Figure 7.5 shows the algorithms for the first three stages of the proposed object definition over the object life cycle. Figure 7.6 illustrates this concept using an example of a wall object. In these two figures, the usage of the object is demonstrated over two construction applications i.e. specifications and construction planning. This section, however, explains the object definition concept for the wall object using the elements' specification application only.

Following the ICE modularised structure, the Specification application is an independent data module which is responsible for generating specifications for building elements created and storing them within its data module. It also provides services such as the sharing of data with the design module, the construction planning module, and estimating module. Through this application, users can alter the specification of any building element at any stage of the building life cycle and examine its impact on other construction applications.

In SPACE, construction project specific information is normally entered through a CAD package and then transferred dynamically to the project model. When a wall is created in AutoCAD, a representative object is automatically created in the CAD environment [Ewen & Alshawi, 1993]. This object encapsulates all the necessary information, which defines its basic features, i.e. the first group of information. In the case of a cavity wall object these are: rotation angle; associated building elements (objects which are built into the wall such windows and doors); start co-ordinates; level; X, Y, Z dimensions; cavity thickness; inner leaf thickness; and outer leaf thickness. Each of these item of data has a role to play in serving the rest of the applications. At this stage, the basic definition of the wall is established and the object is transferred to the project model where it enters the second phase of its life cycle.

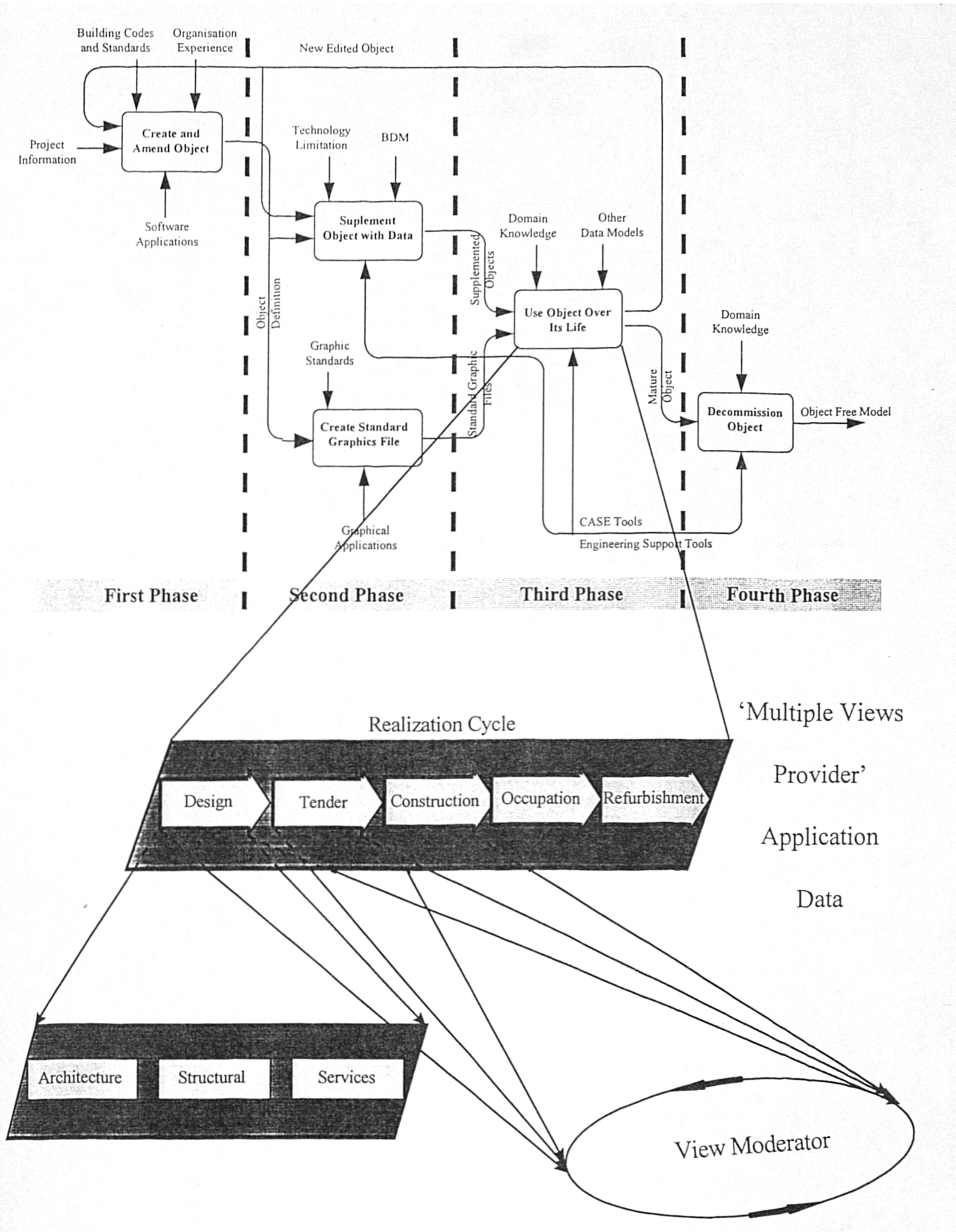


Figure 7.4: The concept of providing multiple views

When the wall object is passed on to the project model, an instance is created for this object and is attached to the design data module under the class cavity walls. The basic definition features, which have been defined earlier, are stored in the object along with those inherited from the cavity wall class. The latter includes data such as object identity (member of the cavity walls class), its construction activities, associates type (type of relationship that the cavity wall has with other objects such as supported by, embedded in, etc.), the European Construction Index the SfB code (Samarbetskommittén för Byggnadsfrågor) [CI/SfB, 1976], the CPI code (Co-ordinated Project Information) [Building Project Information Committee, 1987], and SMM7 code (Standard Methods of Measurements) [SMM7, 1991]. The SfB code identifies the functional attributes of the object, the CPI code represents the work sections according to the CPI CAWS (Common Arrangement of Work Sections) proposals, and SMM7 code describes the object in terms of its work content. These codes assist other applications to classify building elements in various formats. For example, a wall can be part of the brickwork elements, super structure, envelope, etc. At this stage, the object has established an independent status, which is recognised by the ICE and in turn by all other applications.

The wall instance then requests its geometrical information to be generated by the CAD package and to be stored under its identity name in a file. The wall object then refers to this file when it needs to draw itself. At the same time, the wall requests the Specification application to generate an appropriate specification for it. This will trigger off the specification application which first checks whether an identical wall exists. If so, the Specification application asks the wall object to share the existing specification of similar walls. Otherwise, the Specification application:

- retrieves the wall's basic features from the wall object;
- matches these feature with those available in the application data base;
- retrieves those specifications which best match the features of the wall;
- creates a specification object in Specification data module;
- sends a reference to the wall object.

When the wall is deleted, its instance is deleted from the design data module as well as its related specification object if it was the only wall type that refers to that specification object.

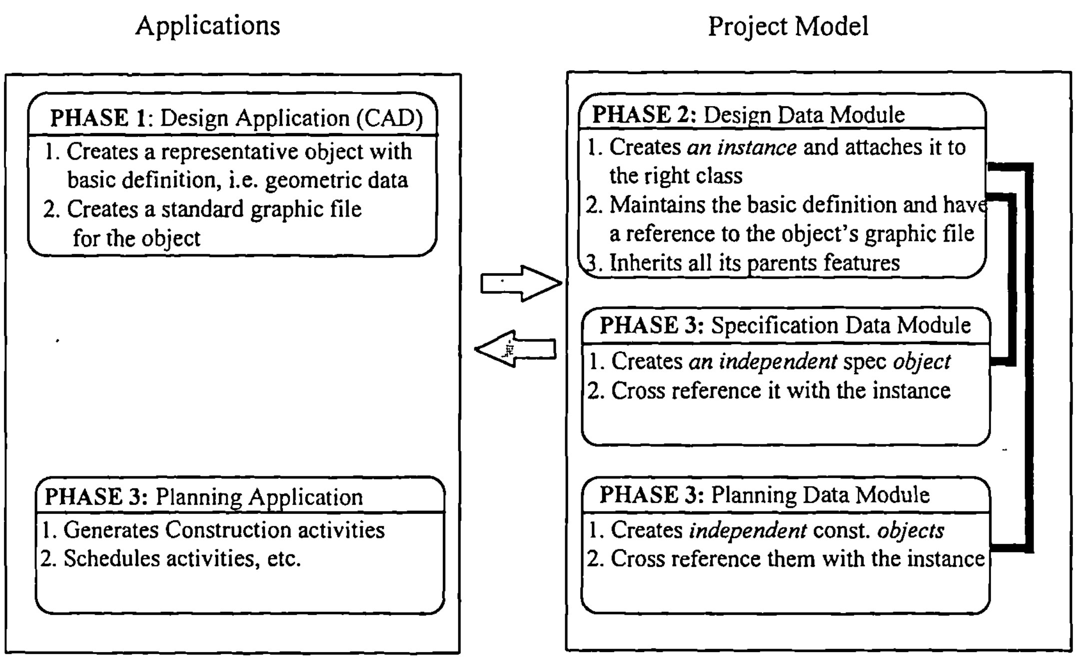


Figure 7.5: The Algorithms of the first three stages of the proposed object definition over the life cycle

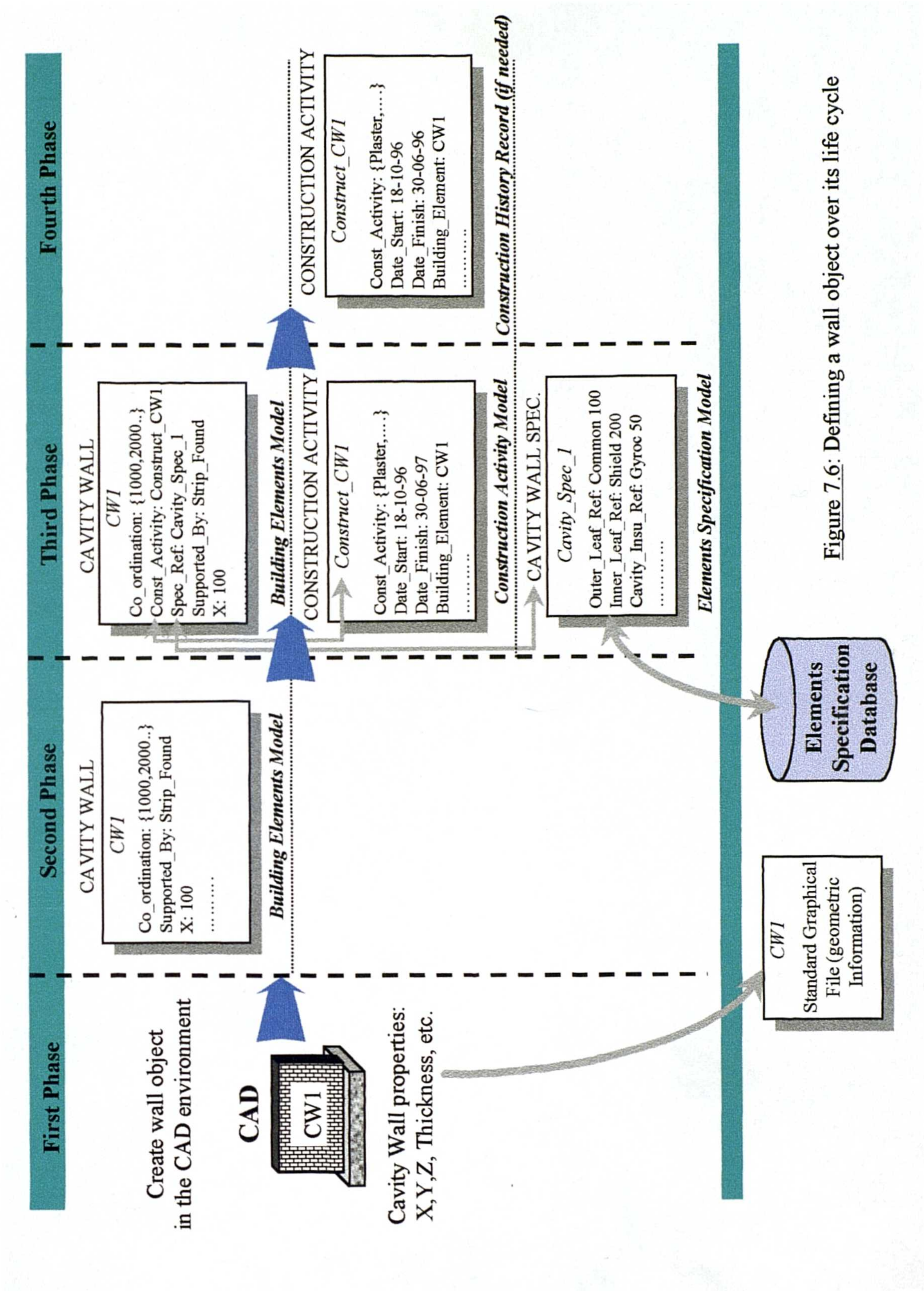


Figure 7.6: Defining a wall object over its life cycle

7.9 Summary

Integrated Design and Construction has been the main issue in Computer Integrated Construction (CIC). It has been widely recognised that in order to achieve the best performance, such integrated systems need to share data via a central core, i.e. a product model. Great efforts therefore have been devoted towards the development of a flexible and comprehensive data structure with the aim of serving all possible downstream applications. However, central cores cannot be populated with project specific information without clearly defined objects, which are anticipated to be transferred dynamically to the central core.

In this context, the STEP APs (Application Protocols) are considered to be an important step towards the identification of application specific data models. However, the development of such protocols in isolation of any implementation attempt will lead to the development of theoretical models which could prove difficult to implement as implementation problems could be of a completely different nature. Thus, it is important that an agreed structure for an integrated environment be establish in parallel to the development of the data models. This can significantly improve the implementation process and can yield an important feedback on the developed application protocols. This chapter has addressed the problems with the current data models, from the implementation point of view, and concentrated on two main issues; object's definition and object's life cycle.

This chapter has discussed the definition of object's within the context of the proposed Integrated Construction Environment (ICE) within which a structured concept for object's life cycle is also introduced. At the creation stage, i.e. in a CAD application, objects inherit two groups of data; global data and specific data. Global data are attached to all elements as they are created while the specific data are attached to specific building elements. Each single data, which is attached to an object, serves specific needs in the ICE and is stored in the most efficient way. Other data are supplemented to objects when they are transferred into the central core of the ICE. An instance will be created in the product data model for each object created at the design application which then takes the objects into the other stage of the life cycle of the object.

Four phases have been highlighted for the object's life cycle. These are; create and amend, supplement object with data, use object, and decommission object. Objects are populated with data at various phases of their life cycle. After instantiation, i.e. at the third phase of the life cycle, objects can either take a new status, i.e. when edited, or become mature objects. Objects at that stage can refer to their own data, to data that they have generated in other application data modules, or to common data. At the end of their life, they can either be deleted from the project model or become obsolete.

Chapter 8

CAPE: The Building Elements Data Module

8.1 Introduction

The preceding chapters discussed the issues of product model and its importance in the development of integrated environments. Such environments require project information to be well structured and classified, represented in the form of models so that it can be identified and understood, and possibly agreed upon in the early stage of the integrated environment development.

The information model, which represents all the different views about the building, requires considerable inputs of logical and practical analysis in its design [Riley & Sabet, 1994]. This information model should cover the facts of the products and processes needed to construct these products. Riley and Sabet [1994] have highlighted that information models should fulfil the following requirements:-

- *General usefulness* – many different applications will use the models for performing different functions;
- *Richness* – the models must capture significant levels of details
- *Comprehensiveness* – the scope must include a wide breadth of project information
- *Flexibility* – is required both in what information can be represented (e.g. numeric parameters, logical relationships, etc.) and in how the information can be used.

As discussed earlier in Chapter 6, *CAPE* is a design application, which includes two main parts, one of which represents a building element data module. *CAPE* is designed to serve all the application modules in the project model. It provides all the necessary design information such as co-ordination, dimension, etc. which are required by the downstream applications. The developed models represent a framework for the presentation of design information, with the aim of establishing a computerised tool to assist both designers and construction professionals such as project manager. It also aims at establishing a central core for the project model to facilitate the integration of design and construction.

This chapter discusses the process of modelling the information required for developing a building element data module using the EXPRESS-G modelling technique. In the earlier part of the chapter, the EXPRESS-G models and notations are briefly presented. This is then followed by the phases involved in developing the EXPRESS-G data models. The developed models are depicted and discussed, firstly in the form of the context diagram, and followed by the level 1 and 2 diagrams.

8.2 EXPRESS-G models and notations

EXPRESS-G is the graphical representation of EXPRESS (an information modelling which consists of computer-readable data definition language [Schenck, 1989]), using graphical symbols to form a diagram [Eastman, 1994]. The EXPRESS language, is one of the technologies that have been developed as part of the STEP standard for product data exchange [Schenck & Wilson, 1994], which represents the information models in textual format. However, the introduction of EXPRESS-G, has replaced the representation of the graphical format whereby the symbols are assembled together to form a diagram.

8.2.1 Graphical symbols

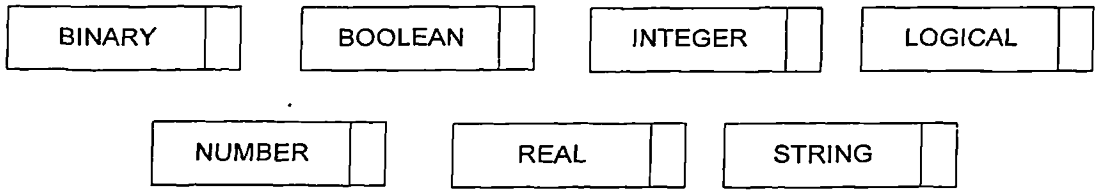
There are three kinds of basic symbols defined in EXPRESS-G: definition, relation, and composition. The contents of the structure of an information model are being defined using the first two symbols, i.e. definition and relation while the third symbol enables the EXPRESS-G diagram to be spread across another physical pages. The following sections describe in detail all the symbols mentioned above.

(a) Definition symbols

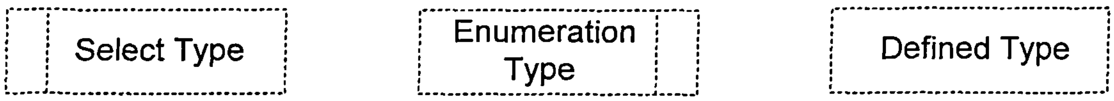
The *definition* symbol denotes the things (i.e. ideas, concept, etc.) which form the basis of the information model. There are four types of definition symbols in EXPRESS-G; entity, schema, simple type and data type. Each of them is represented by a rectangular box with the name of the item being defined. The type of definition is signified by the style of the rectangle. Figure 8.1 (a) depicts the symbol of an entity, a modelling construct that represents some form of item of interest in the real world. Figure 8.1 (b) depicts the symbol of a schema, which is a collection of items forming part of, or the whole model. It is represented by a solid rectangle divided into half horizontally and the name of the schema is written in the upper half of the rectangle, with the lower half being empty. These schemas are used to allow partitioning of data onto discrete topics of interest.

Figure 8.1: Symbols for an entity and a schema

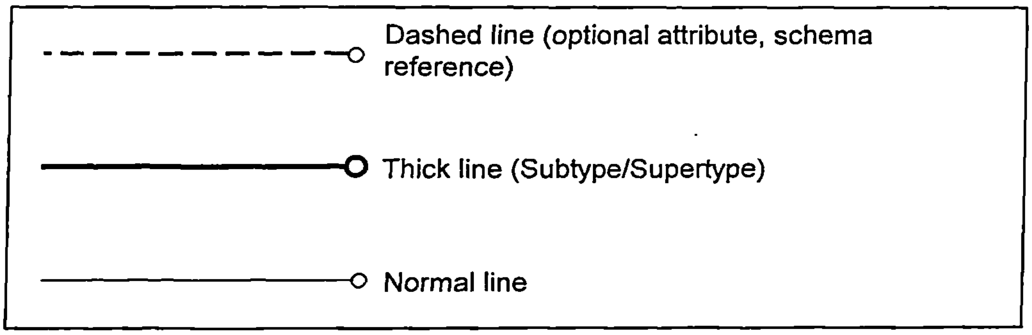
An entity may have attributes, which are the specific properties of interest, i.e. properties that identify an interesting trait and how it is represented. These properties may be either of simple types or complex types (other entity) which are supported by EXPRESS-G . The simple types include Binary, Boolean, Integer, Logical, Number, Real and String. The symbol for the simple type is a solid rectangle with double vertical line at its right end with the name enclosed within the box, as shown in Figure 8.2 (a). There are also three types of symbols used, namely, select, enumerate and define data types which are represented as dashed boxes as shown in Figure 8.2 (b).



<u>Figure 8.2 (a)</u>: Simple type symbols



<u>Figure 8.2 (b)</u>: Type definition symbols



<u>Figure 8.3 (a)</u>: Relationship line styles

(b) Relationship symbols

A *relationship* can be defined as an association between two constructs in a model [Schenck & Wilson, 1994]. A relationship may be implicit or explicitly identified, for example, there is an implicit relationship between an entity and its attributes. The relationship symbol is represented by a line, which connects the definition, and composition symbols denoting relationships between the defined items. There are three different types of different line styles; dashed, thick or normal, as shown in Figure 8.3 (a). A dashed line represents a relationship for an optional valued attribute. A thick line represents a subtype/supertype relationship. A subtype is a kind of specialisation of its supertype, and, conversely, a supertype is a generalisation of its subtypes. Finally, a normal line represents all other relationships.

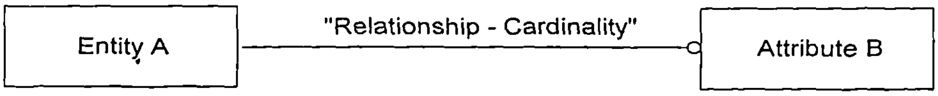


Figure 8.3 (b): Open circle showing the *to* end relationship

Relationships are bi-directional, with one direction being emphasised. For example, if an entity A has an explicit attribute whose type is B, then the emphasis will be in the direction A to B. In EXPRESS-G, the notation of *to* end of the relationship is marked with an open circle as shown in Figure 8.3 (b). Aggregate data types in relationships may be indicated by abbreviations such as L followed by the cardinality constraints. *Cardinality* refers to the specification of the number of instances of one construct that can be associated with one instance of a related construct. In some cases, the inverse relationship may be indicated.

(c) Composition symbols

The *composition* symbol enables a model diagram to be displayed on more than one page. This is essential for large model which cannot be fitted into one page for presentation. Boxes with rounded corners are used to represent the composition symbols, as shown in Figure 8.4. For "reference onto this page", "page #" represents the page on which the entity/attribute is currently shown with reference number "ref #", while "#, #, ..." represent the page(s) from which the entity/attribute is referenced. For "reference onto another page", "name" refers to the named entity or attribute which is referred on page "page #", with reference number "ref #".

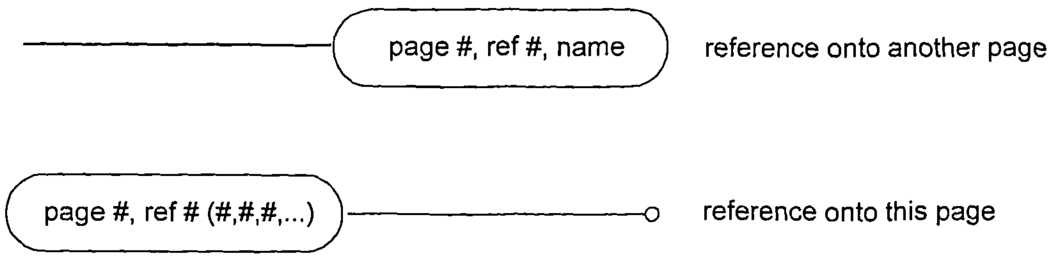


Figure 8.4: Composition symbols

8.3 Developing the EXPRESS-G model

Prior to the development of the data models, the scope has to be specified. The scope of *CAPE* data module comprises all information related to the development and presentation of the building elements data module. However, as *CAPE* is a central core for the integrated construction environment (ICE), the development of *CAPE* data module has cover information which is necessary for other applications such as construction and site planning, estimating, etc. Although the aims of developing *CAPE* data module are to assist the integration of design and construction within integrated construction environment, the scope of the module is limited to the scope of ICE as explained in Chapter 6. Moreover, the level of modelling details was decided upon by considering its dependency on other related information or data modules.

Once the scope of the module has been identified, all the information pertaining to the development of the *CAPE* data module is gathered, organised and classified to allow for a systematic creation of the models. The modelling process followed the proposed structured framework for ICE and the object definition which have been presented in Chapter 6 and 7. Where appropriate or necessary, other relevant sources of information have been referred in order to fill any gaps in the information towards the development of the module.

The starting point for the information modelling was conducted by using the top-down approach, which has been proposed in Chapter 5, i.e. emphasising upon the "abstraction level'. The information was first analysed and modelled at a higher level, followed by the lower levels of abstraction. However, in certain parts of the models, the different levels of abstraction need to be cross-referenced which resulted in a mixture of top-down and bottom-up strategies.

The development of the data module also separates the domain into several models, each describing the information needed to support a single well-defined activity as proposed in Chapter 5. For example, several data models such as the building spaces and the building design data models have been developed and separated from the domain model i.e. the building element data module.

The development of the data module follows the four-phase approach recommended by Schenck and Wilson [1994]. These phases are:

- Basic objects
- Relationships and attributes
- Completion of constraints
- Model integration

8.3.1 Phase 1: Basic objects

The main objective of this phase is to develop the major aspects of the items/objects and the general structure of the model. In other words, this phase involves finding and identifying the real world entities, according to the problem domain or scope, which in this case refers to the entities relevant to the development of the building elements and ICE. This entails recognising the major pieces of information that need to be represented in the model, and selecting appropriate names for these items. This involves, firstly, the creation of entities that will form the data model at the highest possible level, i.e. the Context Diagram. This is followed by the identification of entities at lower levels of detail, labelled as Level 1, Level2, and so on. For example, the entities identified for the context diagram are "Project", "Project Type" and "Project Information", along with the "Brief", which have been obtained from the structured framework of the ICE in Chapter 6.

Once the entities have been identified, the next stage involves finding and identifying their attributes. By describing the major aspects or properties of the entities, simple type attributes are identified. During the process, more complex attributes also emerged, which require further objects for their representation. This results in the identification of a new object, which is then added to the model. For example, "Beam Design" was initially identified as an attribute of "Structural Design Elements". With further analysis, it was discovered that this attribute is in itself a large source of information, which can be further sub-divided. This leads to the formation of the "Beam Design" entity.

The next step is to find out whether the entities can be categorised and whether there are any specialisation relationships between the entities. This involves recognising that some entities may be of a special type of another entity, which will result in a supertype/subtype relationship. This normally leads to the identification of the more general entity, i.e. more general name for the supertype, which can then be added to the model. For instance, the subtypes added to the "Project Type" entity were "Roads", "Buildings", "Bridges" and "Others" as shown in Figure 8.6.

The entities and their attributes are then documented by describing the intended meanings of all the constructs in the model. This is followed by the examination of the simple types to ensure that they are appropriately used. At the end of the phase, the model is reviewed to ensure that the developed entities and attributes represent all the information required by the ICE and it's building elements data model.

8.3.2 Phase 2: Relationships and attributes

This phase is concerned with refining the model developed in phase 1. It entails the identification of the relationships between the entities in the model. The behaviour in which the entities are associated with each other is determined, and the constraints on these relationships, if any, are identified.

The process of categorising the entities is repeated in this phase. The entities are categorised to identify any inheritance, subset or specialisation relationships between them. Further categorisation takes place, especially at the lower levels of detail.

Throughout the phase, the categorisation structures in the model are reviewed and refined.

This phase also calls for re-examination of simple types to check whether they are appropriate. Entities and attributes are also examined to find out whether additional attributes are required to characterise an entity. The method of deriving the attribute values is identified to see if these values are dependent on or derived from other attribute values. If so, these are recorded and the necessary relationship added to the model.

Further analysis includes checking whether the existence of one entity is dependent on its usage by another entity. Indirectly, this leads to a form of relationship being identified between the two entities. Combinations of attribute values are also determined to see if any combinations uniquely identify an entity instance. To complete the relationship identification, local consistency constraints are identified to see if they are applicable to the entities. An example of a local constraint is where the value of an attribute is constrained within a certain range, such as the number of months in a year is between 1 to 12.

During this phase, entities, relationships, attributes, and types are added to the entity-level models. This phase is iterated at various levels of abstractions until the model reaches the desired level of details. The model is then reviewed to ensure that all of the relevant information has been embodied. Finally, the resulting entities, attributes and relationships are documented.

8.3.3 Phase 3: Completion and constraints

Once the developed model is sufficiently mature and stable, the global constraints are defined. Global consistency rules refer to constraints that either apply between entities in a model, or to some instances of a particular entity. When the global rules are identified, they are documented. During this phase, further analysis is carried out to ensure that all the existence dependencies, the uniqueness constraints and other cardinality constraints are captured. Existence dependency refers to the situation where the existence of an instance of a class is dependent on an instance of another related class. A uniqueness constraint, on the other hand, occurs when the value of some attributes, or a combination of attribute values, are unique across all instances of a class, which indicates that they can be used to identify objects. The model is also checked to see if there are any local consistencies rules.

Another consideration is to whether the model developed is well partitioned. The partitioning of the models was carried out based on the levels of detail of the information required for the development of the ICE and the building elements. Entities, which require further examination, were partitioned to ensure a rigorous portray of the contents of the ICE and the building elements at various levels of details, as well as, for understanding and explanatory purposes. This gave rise to the models being developed at different levels of diagram; Level 0 (context diagram), Level 1, Level 2, and so on. At the end of this phase all the constraints are documented.

8.3.4 Phase 4: Model integration

Phase 4 is usually carried out for a large information model, where the work is broken down into parts with different modelling groups being responsible for certain parts of the model only. When all parts of the model have been completed, they are integrated to form one whole model. For this study, since the model has been divided into partitions at different levels of details, it is essential to make sure that all the components that make up the partitions, at various levels, are consistent, compatible and non-contradictory. Due to the large nature of the model, the overall integration requires more than one physical page. Links or relationships between a component of one diagram (partition) and a component in another diagram, are made by using the composition symbols. Once the overall integration has been carried out, the links between the relevant components are reviewed to ensure consistency in the data model.

This phase represents the final stage of the model development process. The resulting data model portrays the full representation of the information required for the development and presentation of the building elements.

The succeeding sections in this chapter present the developed building element data module for the ICE. The constituent models are presented sequentially based on the levels of detail within them. It begins with the highest level, i.e. the context diagram, followed by the Level 1 and Level 2 diagrams. For each diagram, the main components of each model are described. Where necessary, the source from which the component of the model has been derived is also highlighted.

To assist with the presentation and discussion of the data models in the following sections, each data model is depicted by a figure, for example Figure 8.5. Each figure is also labelled as "Page a of b", where "a" refers to the page corresponding to the data model; and "b" refers to the total number of pages for all the data models, which is 10. The page reference denoted by the composition symbols on the data models refer to the above mentioned data model pages, and NOT the pages of this thesis.

8.4 The context diagram – High level view of the project presentation

Figure 8.5 depicts the highest entity-level model of the ICE presentation whereby the information constitutes in this level portray the integration of design and construction over the project life cycle. It illustrates that in a "Project" which has been described by "Brief", may contain the "Project Information" and "Project Type" in which "Project Information" determines the development of a "Project". Conversely, a "Project.Type" is subjected to a "Project" which is described by the client "Brief", i.e. whether it is a building, road, etc. On the other hand, "Project Information" may be influenced by one or many "Project Type".

A composition symbol is attached to the project information and the project type entities which indicates a reference to other pages. For example, the composition symbol for the "Project Type" entity shows the number "1, 1 (2)". The first number "1" refers to the data model page, Page 1 (Figure 8.5). While the second number "1" is the reference number attached to the "Project Type" entity, and the number in the bracket refers to the page number(s) of the model from which further information on "Project

Type" can be obtained, i.e. Page 2 (Figure 8.6).

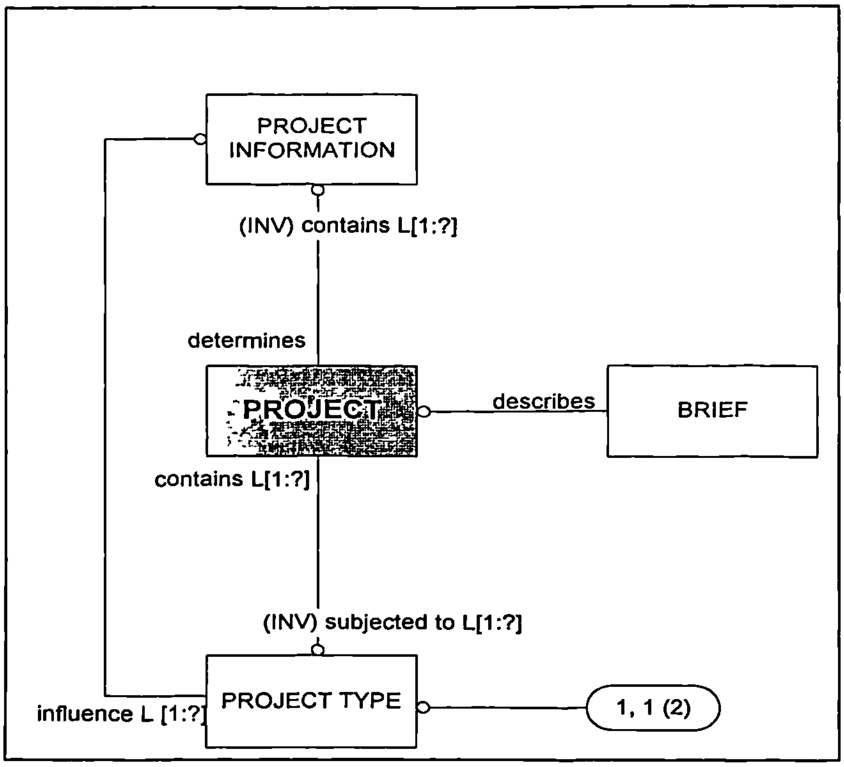


Figure 8.5: Context diagram – High level view of the project information (page 1 of 10)

8.5 The Level 1 diagrams

At this level of abstraction, the main entities which form part of the "Project" entity depicted in Figure 8.5, are examined and decomposed further. Each main entity is presented individually or otherwise, along with their attributes and subtypes, where applicable.

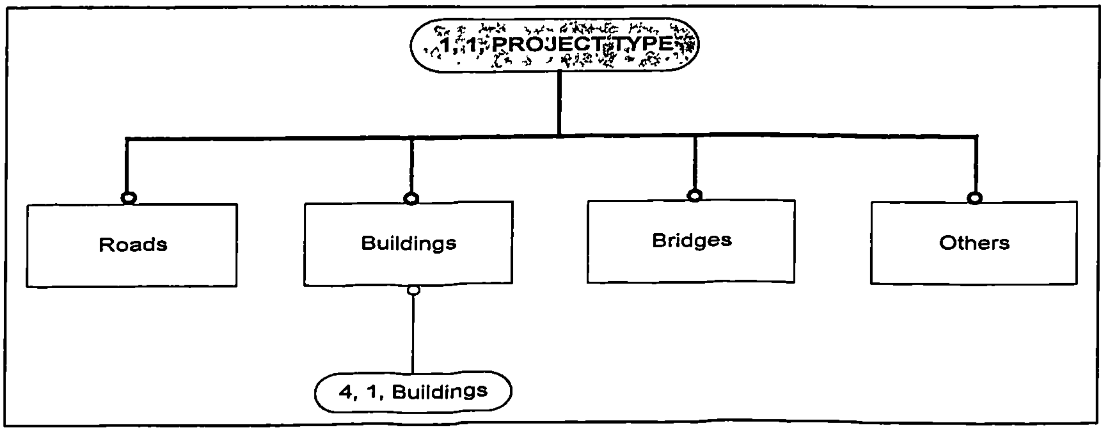


Figure 8.6: Level 1 diagram – Project Type (page 2 of 10)

8.5.1 The "Project Type" entity

Figure 8.6 illustrates the entities of the "Project Type" i.e. "Roads", "Buildings", "Bridges" and "Others". This study is concerned with "Buildings" only. Other entities are depicted in Figure 8.6 to show how other "Project Type" such as "Bridges" could be incorporated as part of the ICE in future expansion.

The other entities linked to "Buildings" are also linked to their respective related entities by the composition symbols. This indicates that the necessary information for these items is obtained from other parts of the model whereby the information pertaining to the "Buildings" were further decomposed. This will illustrates the importance of "Project Type" in the development of an ICE.

8.5.2 The "Buildings" entity

The main entity in the *CAPE* data module is the "Buildings" entity. It contains information related to a building project as part of "Project Type". It composed of one or many "Building Type", "Building Space", "Building Elements", "Building Design" and "Building Other Specifications" as shown in Figure 8.7. The "Building Type" contains three subtypes, i.e. "Office", "House" and "Others". This is to illustrate that the information related to a specific building type can be stored in this entity. For example, the knowledge of construction method, design method, etc. of a particular office type can be stored in the "Office" entity. The "Building Space", "Building Elements", "Building Design" and "Building Other Specifications" are decomposed further in later diagrams. This diagram also shows that "Building Design" entity can use one or many "Building Elements".

The "Buildings" entity also has one or many attributes such as "Associated Elements", "Co-ordinations", "Rotation Angle" and "Building Boundary". This information is needed to illustrate the basic information about the building. For example, the "Co-ordination" and the "Rotation Angle" provide the location and the orientation of the building which are required by the site planner to allocate access roads, materials storage, construction plant etc. The "Building Boundary" entity provides "Building Space Area" which can be defined by external columns.

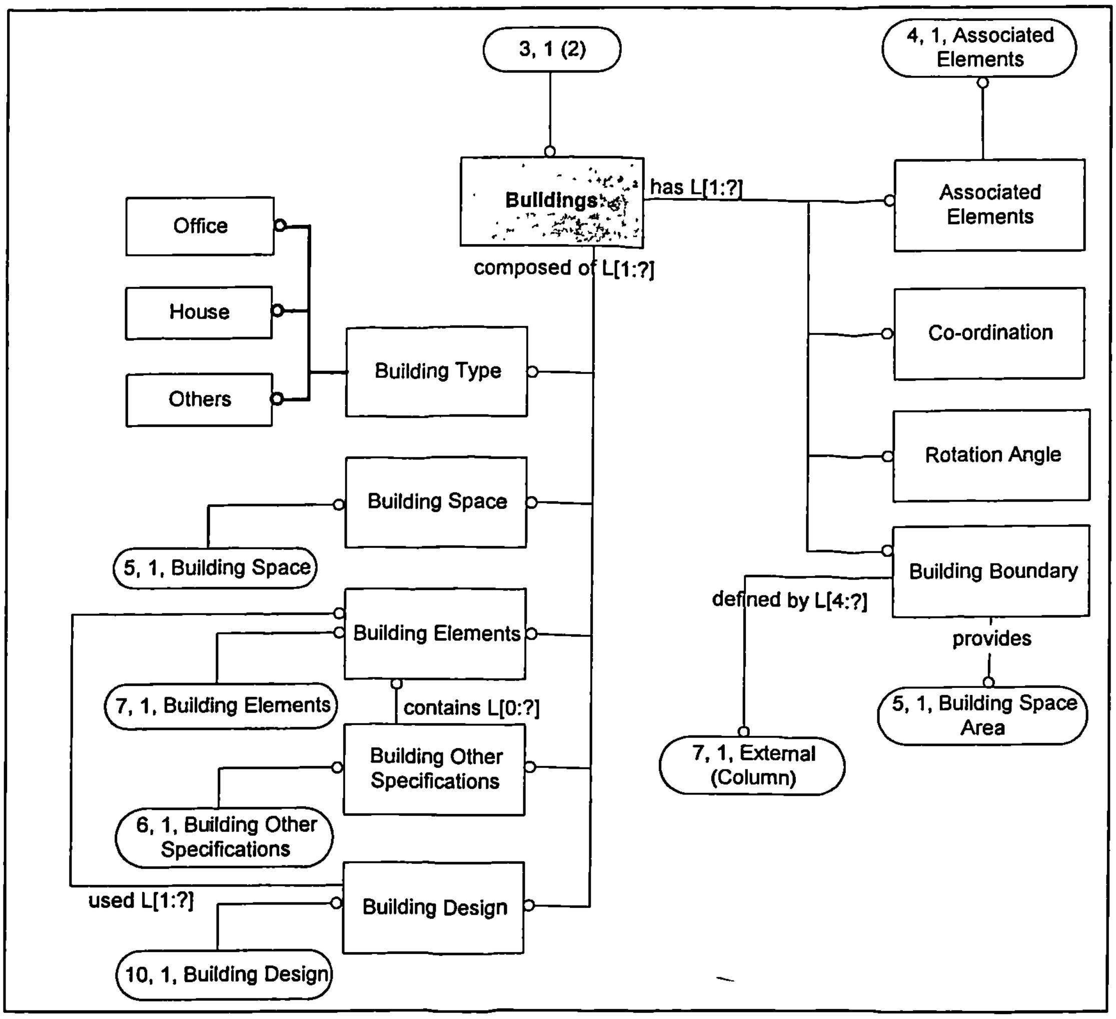


Figure 8.7: Level 1 diagram – Buildings (Page 3 of 10)

The "Associated Elements" attribute shows those building elements which are associated with each other. The "Associated Elements" are defined by at least two or many building elements as shown in Figure 8.8. The attribute used to represent the "Associated Elements" is the "Topological Relationship" which it is represented by at least two or more "Associated Elements". On the other hand, the "Topological Relationship" inversely occurred between at least two or many building elements. For example, a wall must be attached to two columns and a slab can be supported by four beams. The type of "Topological Relationship" can be "embedded in", "attached to" or "supported by" relationship. In the construction planning application, the "Topological Relationship" plays an important role, i.e. the construction activities are influenced by at least one or many types of "Topological Relationship".

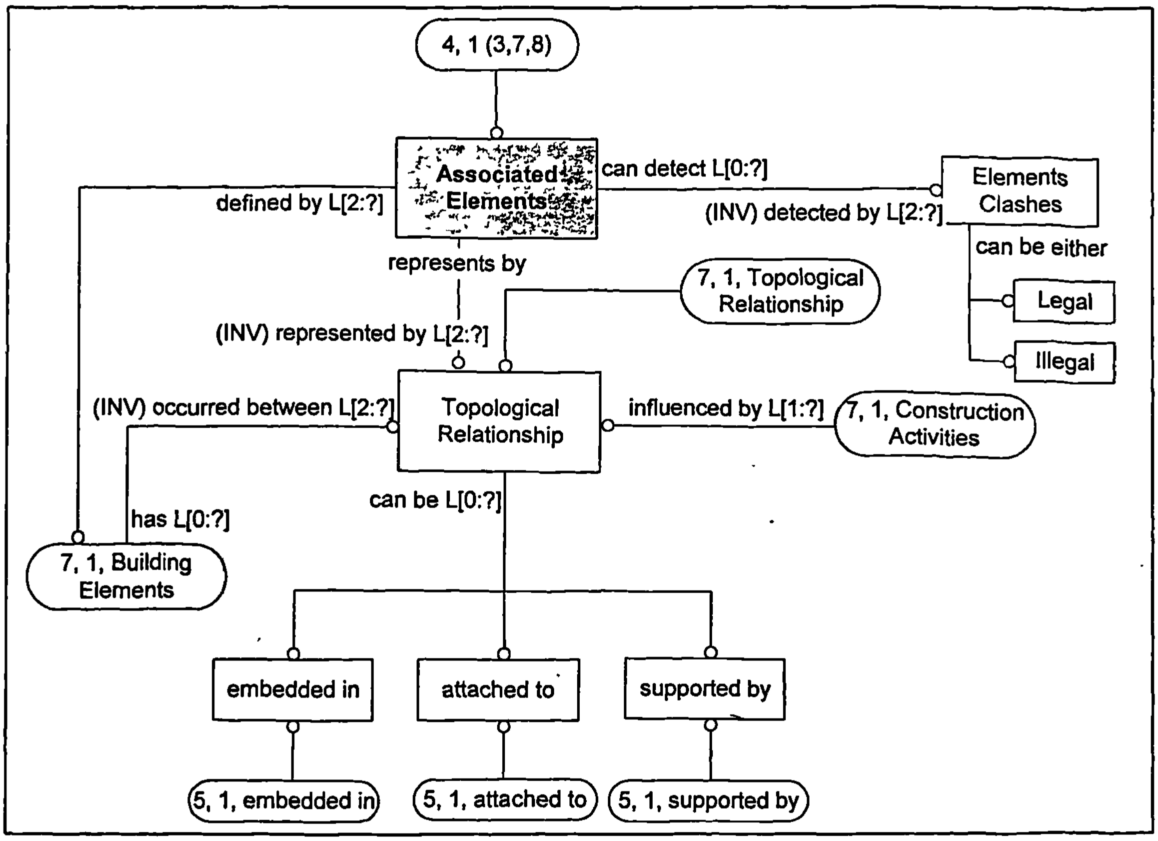


Figure 8.8: Level 1 diagram – Part entity-model of Buildings – Associated Elements (Page 4 of 10)

The other important role of the "Associated Elements" entity is that it can detect whether there are any "Element Clashes" which can be either "Legal" or "Illegal" clashes. For example, a column, which is supported, by another column is a "Legal" clash whereby a ducting system, which intersects a beam, is an "Illegal" clash. Inversely, the "Element Clashes" are detected by at least two or many associated elements.

8.6 The Level 2 diagrams

The "Buildings" main entities; the "Building Space", "Building Other Specifications", "Building Elements" and "Building Design" have been decomposed further in Level 2 diagrams. These diagrams portrayed how the information are shared and exchanged. The following sections explain Level 2 diagrams.

8.6.1 The "Building Space" entity

The central entity in the end user's and the architect's view of the building is the space. A space is a volume bounded on all sides by enclosing structures, which forms the physical boundaries of the space [Björk, 1992b]. The "Building Space" entity is an important entity for space layout planning which is one of the most interesting and difficult design problems. It also plays an important role in the data model because of its relationship with most of the design systems such as architectural, structural and services. This entity contains two entities (Figure 8.9). It contains one or many "Floor Space" which in turn occupy the "Building Space" and provide "Building Space Area" which is needed as one of the parameter of the building. The "Floor Space" contains three types of entities. It contains one or many "Room Space" which built-up or partition-up the "Floor Space". The "Floor Space" could also contain the "Circulation Space" whereby it includes the horizontal (lobby, fire exits, corridor, etc.) and the vertical (lift) type of circulation. The "Floor Space" could also contain "Services Space" which is used by one or many "Services Elements". For example, a ducting systems occupies the top level of each "Floor Space".

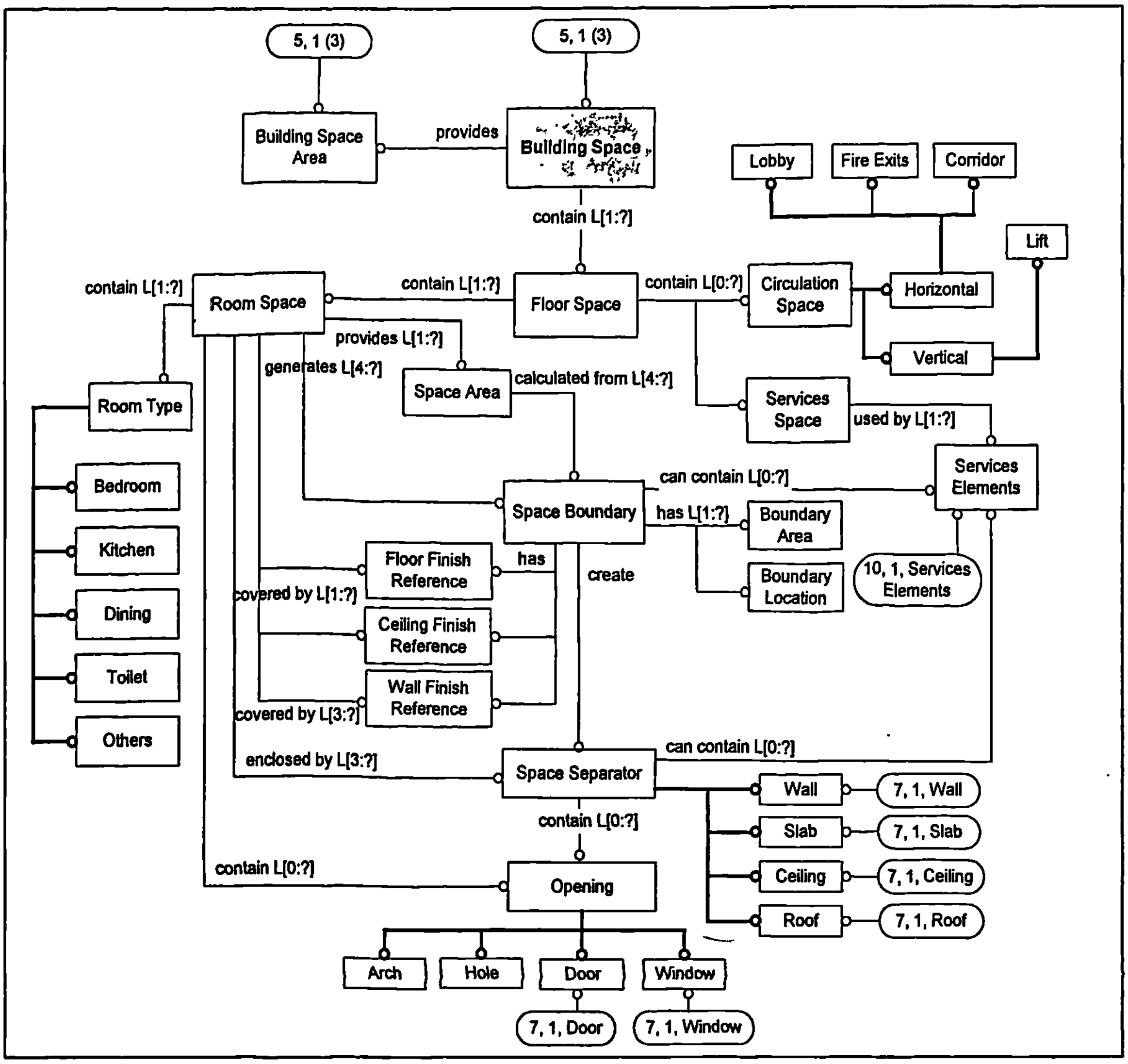


Figure 8.9: Level 2 diagram – Building Space (Page 5 of 10)

The "Room Space" which separates the "Floor Space" contains one or many "Room Type"; "Bedroom", "Kitchen", "Dining", "Toilet" or "Others" which represent the functionality of the "Room Space". This information is very useful as it associates a function to a space. This relationship could be significant in defining the properties of some spaces. For example, specific type of finishing can be associated with toilets, special ventilation systems with conference rooms, etc. It also provides "Space Area" which can be calculated from three or more "Space Boundary". "Space Boundary" is an abstract concept, which represents a part of the infinitesimal thin skin, which surrounds a space or enclosing structures that bound it [Björk, 1992b]. The "Space Boundary" can contain "Services Elements", for example, a radiator which is placed along the boundary. It could also have one or more surface texture or finishes which are referenced, i.e. "Floor Finish Reference", "Ceiling Finish Reference" and "Wall Finish Reference". The "Space Boundary" also has two attributes of "Boundary Location" which can be used to calculate the "Boundary Area". A "Space Boundary" on the other hand, is viewed to create "Space Separator" which enclosed the "Room Space". The "Room Space" is normally enclosed by three or more "Space Separator" and it may or may not contain an "Opening".

The "Space Separator" or enclosing structure is an aggregation of objects which forms the space boundaries of two or more individual spaces (or between spaces and the outside of the building) [Björk, 1992b]. It is viewed as a shell which consists of "Wall", "Slab", "Ceiling" and "Roof" and sometimes with a number of "Opening", which is an abstract generalisation and usually filled with "Window", "Door", "Arch" or left as a "Hole" [Björk, 1992b]. An "Opening" allows the movement of people, light, air fluids, etc. It sometimes contains "Services Elements" such as ducting systems along or in between the "Space Separator", i.e. between ceiling and roof. Thus, it is important in this model to incorporate all the elements together, for example, to calculate the heat lost of a "Room Space", in which the parameter of the "Space Separator" together with the "Opening" are used.

8.6.2 The "Building Other Specifications" entity

The "Building Other Specifications" entity is specially designed to hold the information about the characteristics of the building such as "Level", "Location", "Zone", "Floor" etc. as shown in Figure 8.10. Such information is useful in providing a quick access on a specific information of a building. These entities provide a collection of building elements which have been sorted according to specified specifications, i.e. "Level", "Location", "Zone", "Floor", etc.

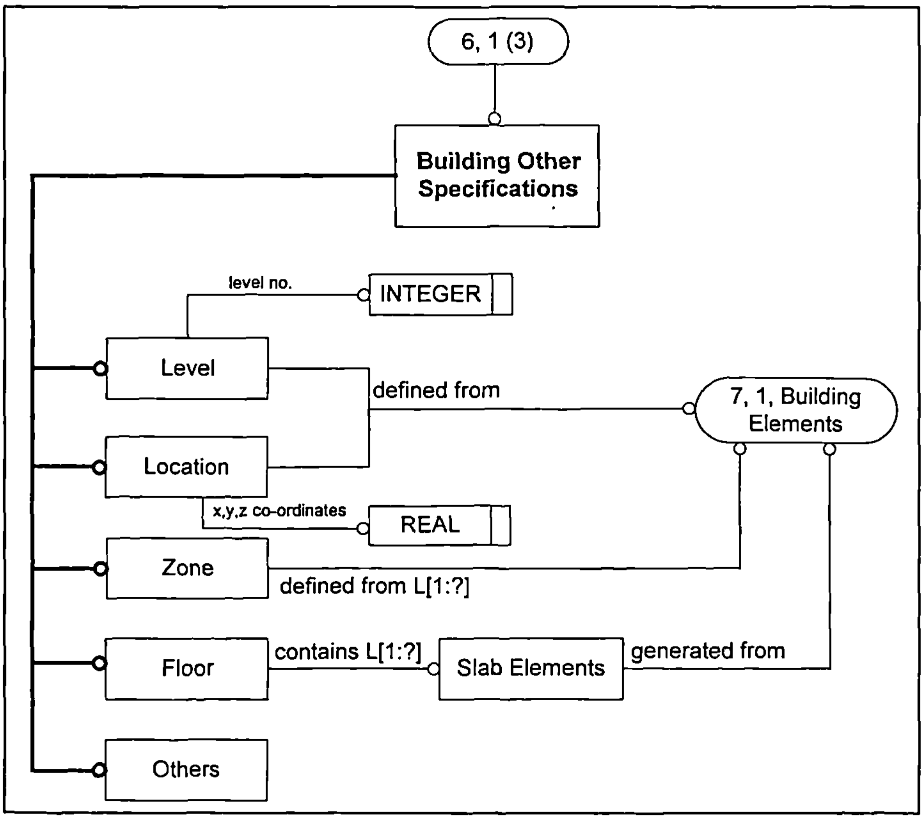


Figure 8.10: Level 2 diagram – Building Other Specifications (Page 6 of 10)

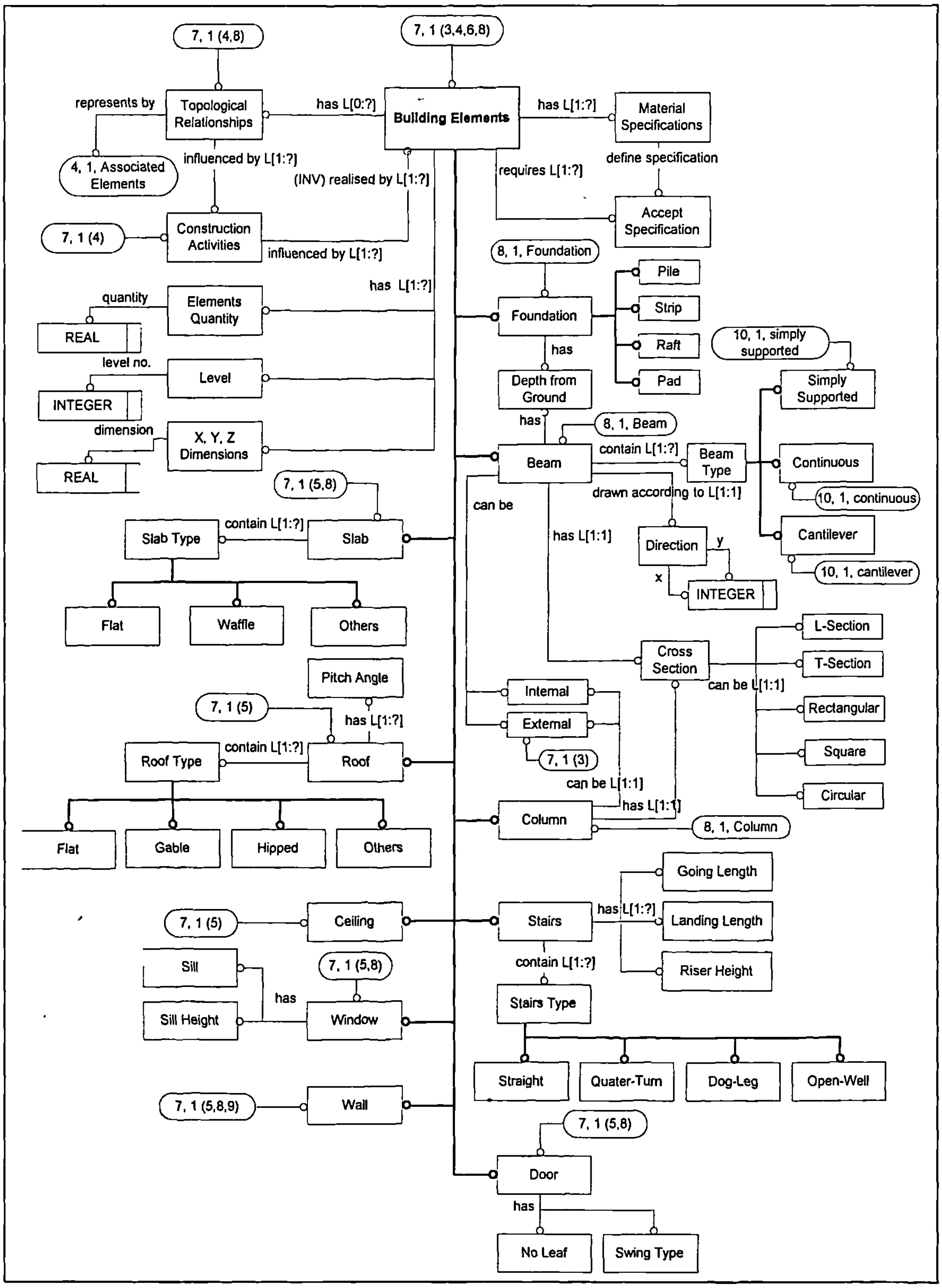


Figure 8.11: Level 2 diagram – Building Elements (Page 7 of 10)

The "Level" and "Location" entities are determined from the building elements where the "Level" is in "Integer" form and the "Location" is in the form of x, y, and z co-ordinates, which is normally in "Real" form. Sometime the building elements are best grouped according to "Zone" e.g. for construction planner to plan their construction activities. Each "Floor" of a building contains one or many "Slab Elements" which built-up the floor levels and is generated by the building elements.

8.6.3 The "Building Elements" entity

The "Building Elements" entity in a concrete framed building can be a "Wall", "Door", "Window", "Ceiling", "Slab", "Foundation", "Roof", "Column", "Beam" or "Stairs" as shown in Figure 8.11.

The "Building Elements" entity consists of several attributes in addition to the main attributes provided for the "Buildings" entity. It includes the "Material Specifications", "Topological Relationships", "Construction Activities", "Elements Quantity", "Level" and "X, Y, Z Dimensions" as shown in Figure 8.11. The "Building Elements" has one or many types of "Material Specifications" such as a "Wall" which is built-up by a class A english type of brick and a mortar mix ratio of 1:3 or a column made from concrete grade of C30 and reinforcement of high yield steel. The "Material Specifications" is defined by the "Accept Specification" which is required by the "Building Elements". The "Building Elements" also has "Elements Quantity" which is calculated from the physical dimensions of the elements, i.e. through their "X, Y, Z Dimensions". The "Building Elements" has one or many "Level" in a building. For instance, a double storey building may have beams at ground and first levels and are numbered as 0 and 1. "Construction Activities" are largely influenced by one or many "Building Elements" whereby inversely, the "Building Elements" is realised by one or many "Construction Activities".

Another important attribute of the "Building Elements" is the "Topological Relationship". The "Topological Relationship" occurred between two or more "Building Elements" and it could be either "supported by", "attached to" and "embedded in" as shown earlier in Figure 8.8. In this study, only these three types of "Topological Relationship" are considered. For example, a "Column" stump is "supported by" a "Foundation", a "Wall" is "attached to" a "Column", a "Door" or "Window" is "embedded in" a "Wall", etc. (Figure 8.12). This information can be very useful in controlling the sequential activities required to construct the "Building Elements". For example, a column which is "supported by" other column must be constructed before the other column can be constructed. Or during the maintenance work, the "Building Elements" which has the "Topological Relationship" with other "Building Elements" must be considered before the maintenance work begins in order to avoid any destruction or failure of the maintenance programme.

The "Building Element" entity such as "Wall", "Beam", "Column", etc. in the model represents the geometric view of the "Building Elements". For instance, a "Column" or "Beam" are viewed according to their shape or "Cross Section" such as "Square", "Circular", "Rectangular", "T-Section", etc. and not according to their design function i.e. slender column, simply supported beam, etc. The design function is presented in a separate model (Figure 8.14). This view makes the "Building Elements" entity as a global entity where it can be accessed and shared not only for the design purposes but also for planning, estimating, etc.

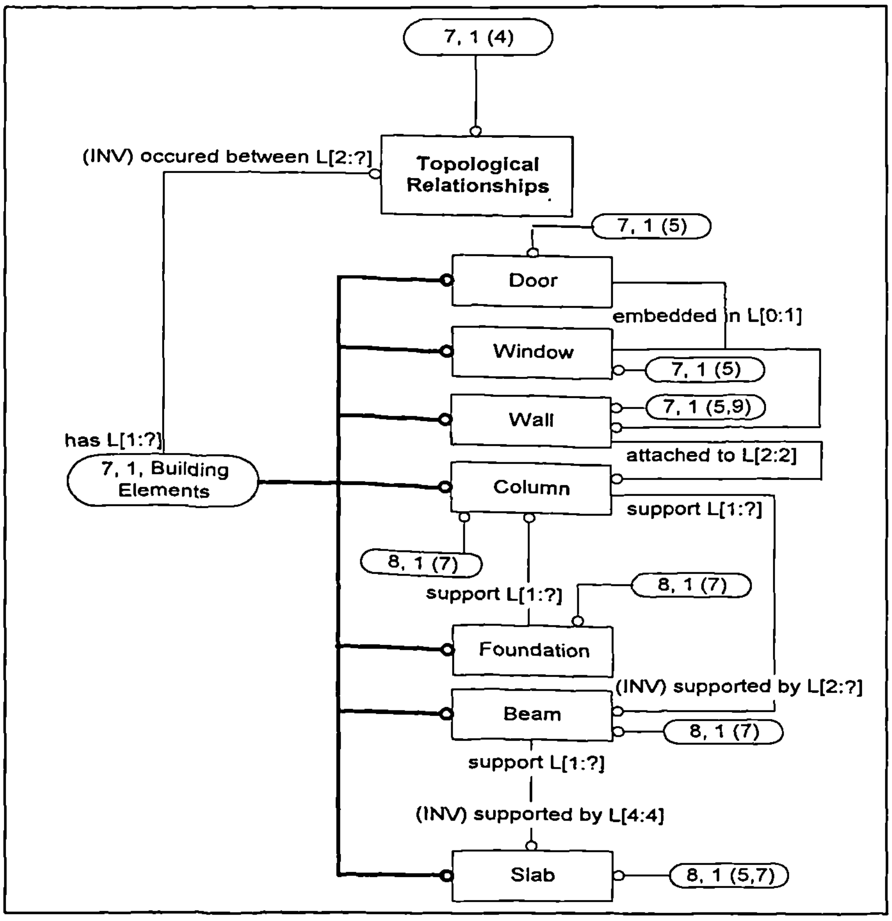


Figure 8.12: Level 2 Diagram – Part entity-model of Building Elements – Topological Relationships (Page 8 of 10)

The "Foundation" entity can be "Pile", "Strip", "Raft" and "Pad" and are constructed below the ground level. Therefore, the "Foundation" together with the ground "Beam" require an extra attribute "Depth from Ground" in order to store the right depth for the excavation works. The "Beam" is a horizontal bearing structure usually made of concrete, steel or wood [Björk, 1992b] contains one or many "Beam Type" of either "Simply Supported", "Continuous" or "Cantilever" which are all drawn according to the "Direction" of x or y. For example, a continuous beam which is constitute more than one "Beam" may not be constructed in one process, i.e. it has to be split up into few parts. In this case, the direction of the second or the remaining "Beam" has to be known in order to define the structural function of the beam i.e. continuous beam. Other "Building Elements" such as "Roof" and "Stairs" entities require specific attributes due to their complex shapes. For example, a staircase require "Landing Length", "Going Length" and "Riser Length" to calculate the quantities of different types of materials required to construct it. The "Door" entity is an opening which is intended primarily for access with hinged, pivoted or sliding operation [ISO 6707-1, 1989]. It has "Number of Leaf" and "Swing Type" attributes to differentiate the types of "Door". The "Window" entity has "Sill" and "Sill Height" which are required for the design and construction purposes.

A "Wall" entity is a vertical member, which bounds or subdivides a building and fulfils a load bearing or retaining functions [ISO 6707-1, 1989]. It has its own unique attributes such as "Structural Functions" which in turn, can be either "Structural" or "Non-Structural" i.e. load bearing or non load bearing. A "Wall" entity also has a "Direction" attribute of x or y direction to give the orientation of the wall as shown in Figure 8.13. A "Wall" entity also consists of several types of wall such as "Cavity Wall", "Partition Wall" and "Solid Wall" which can have their own specific attributes. A "Partition Wall" must be used as an "Internal" wall where a "Solid Wall" can be used either as an "Internal" or "External" but in either case it must be a "Non-Structural" wall. A "Cavity Wall" on the other hand, contains more extra attributes such as "Cavity Thickness", "Outer Leaf" and "Inner Leaf". It can be designed as a "Non-Structural" or a "Structural" type of wall and can only be used as an "External" wall.

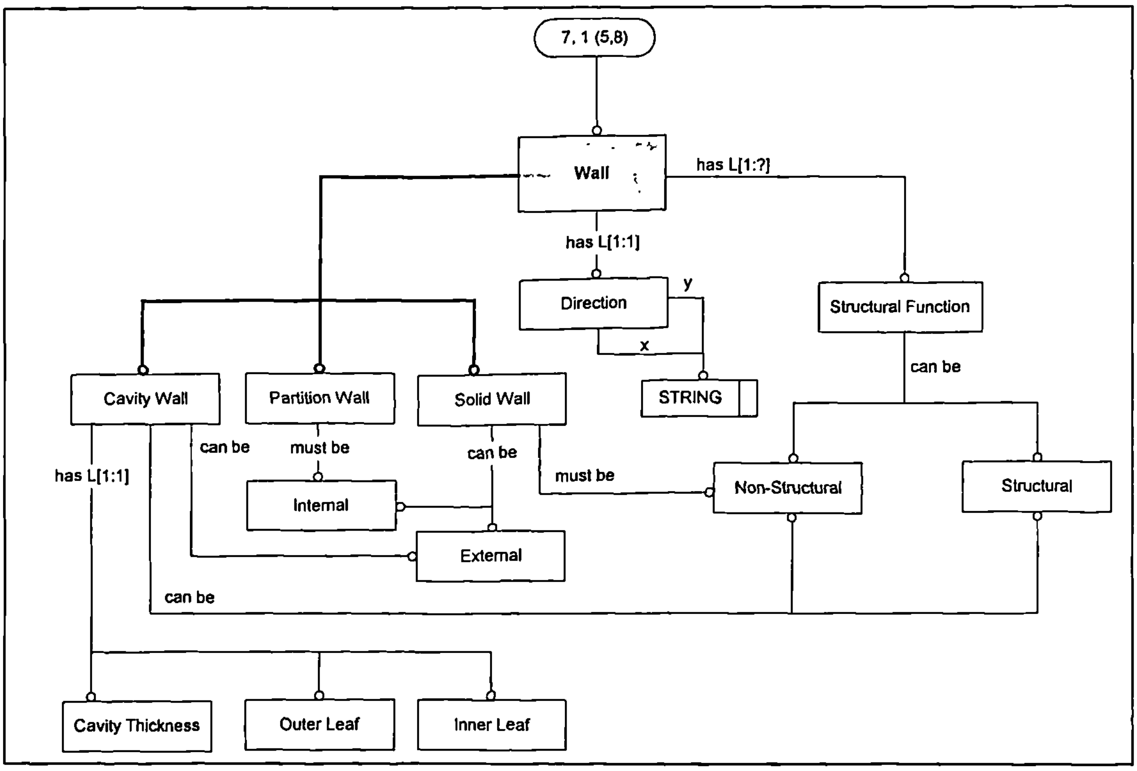


Figure 8.13: Level 2 diagram – Part entity-model of Building Elements – Wall (Page 9 of 10)

8.6.4 The "Building Design" entity

As mentioned earlier in section 8.6.3, the "Building Elements" entity represents the basic views of the elements which is essential to support the different views of an element. For example, a structural engineer is interested to look at a "Beam" as either a "Simply Supported" or "Continuous" whereas a contractor is interested to look at how to construct a "Beam". In the case of a continuous beam with many "Span", which cannot be constructed in one activity, detailed information has to be provided to satisfy the design and construction processes. For such problems, a "Building Design" entity has been created as one of the "Building" entity as mentioned earlier. The "Building Design" entity is composed of two entities, i.e. "Structural Design" and "Services

Design" as shown in Figure 8.14. The "Services Design" includes "Services Design Elements" including "HVAC" and "Others". The discussion or the expansion of the "Services Design" entity is outside the scope of this study. The "Structural Design" on the other hand, includes one or many "Structural Design Elements" such as "Slab Design", "Column Design" and "Beam Design".

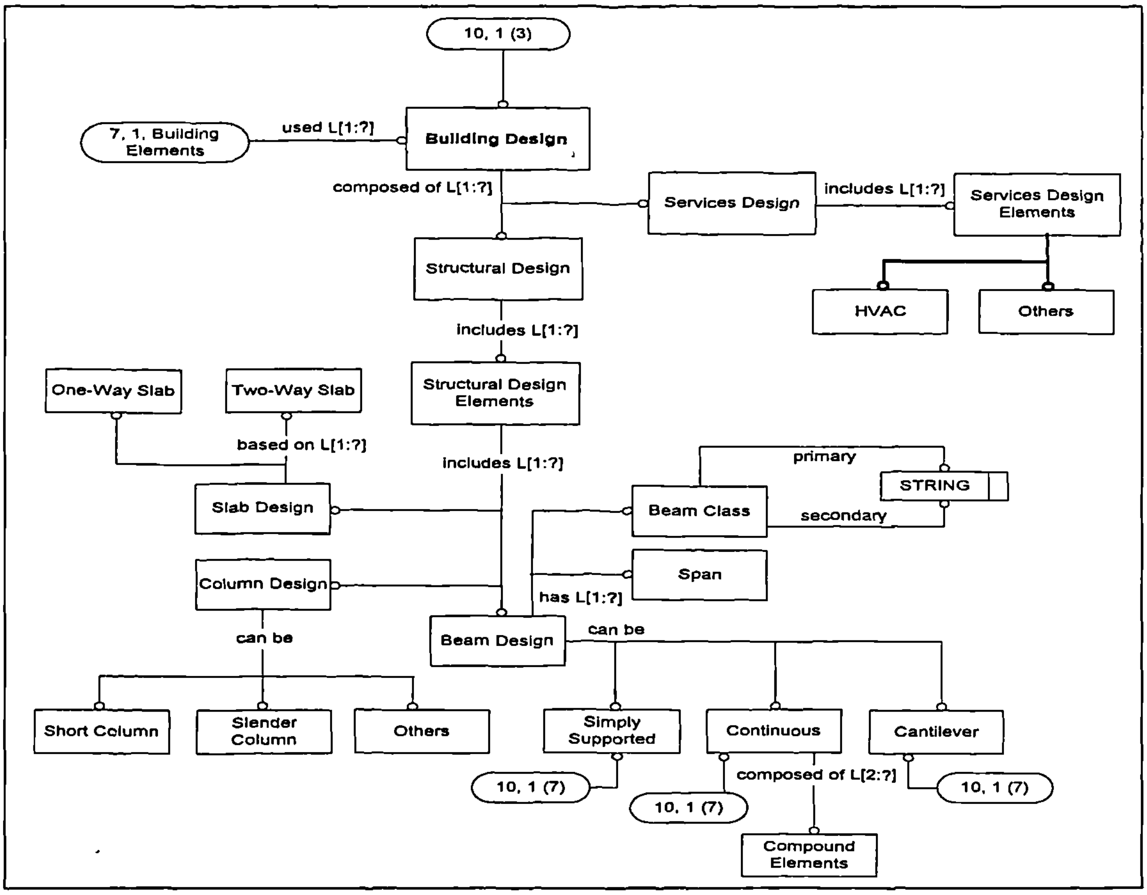


Figure 8.14: Level 2 diagram – Building Design (Page 10 of 10)

The difference between the "Slab" entity in the "Building Elements" model and the "Slab Design" entity in the "Building Design" model is that the "Slab" entity provides basic information on shape or cross-section. While the "Slab Design" entity has been modelled to accommodate the structural engineer's views. The same concept is applied to both "Column Design" and "Beam Design" entities.

The "Column Design" has attributes to define its structural functions i.e. "Short Column", "Slender Column" and "Others" whereby a specific design methods for each type can be stored. The "Beam Design" has one or many "Span" and "Beam Class" where "Beam Class" attribute has a "String" value which can be either "primary" or "secondary". This information is essential for the structural engineer to differentiate the loading values imposed on the beam for the analysis and design purposes. The "Beam Design" can also be a "Simply Supported", "Continuous" or "Cantilever" whereby a "Continuous" beam is composed of either two or more "Compound Elements". The "Compound Elements" attribute may contain a collection of beams, i.e. a group of "Simply Supported" beam. For example, when a continuous beam with three spans has to be drawn, three steps are needed, i.e. three simply supported beams need to be drawn sequentially and continuously to create the "Compound Elements". The rationale behind this, in this data model as mentioned earlier, is to split the element in order to support both design and construction views.

Information for the structural design of the "Building Elements" can be extracted from the "Building Design" entity where all the design information is stored. Other information which are required for construction, estimating, and other purposes such as VR, site layout, etc. can be extracted from the "Building" and "Building Elements" data models as discussed earlier.

The following sections explain the relationship between the "Building" and the "Building Elements" data models and the other modules in the ICE.

8.7 Relationship between the "Building Elements" data module and other modules in the ICE

The main aim of "Building Elements" data module is to represent the design elements, their relationship and behaviour, where it can be accessed, in a structured manner, by other construction applications such as construction planning, Virtual Reality (VR), site layout planning, estimating, etc. Currently, "Building Elements" data module supports five other data modules such as EVALUATOR [Underwood & Alshawi, 1997], SPECIFICATION [Underwood & Alshawi, 1997], CONPLAN [Hassan, 1997], INTESITE [Sulaiman, 1997] and CONVERT [Alshawi & Faraj, 1995]. Brief descriptions of these modules are mentioned earlier in section 6.5 of Chapter 6.

As mentioned in the earlier sections, the "Building Elements" data module provides basic information for other applications to share and exchange. Therefore, in the development process, all other applications have been considered. The following sections briefly discuss the relationship between "Building Elements" data module with other modules in the ICE.

(1) Relationship with EVALUATOR

The main objectives of EVALUATOR are to generate an elemental Bill of Quantity (BQ) and monthly interim valuation certificates [Underwood & Alshawi, 1997]. Each of these has been represented in the EVALUATOR module by "Project Estimates" and "Interim Valuation" as shown in Figure 8.15. The "Project Estimate" is composed of one or many "Project Estimate Items" which are determined from the

"Building Elements. Within this model, certain entities and attributes have been shaded or hatched to represent the information for which the EVALUATOR acquires from other modules, as opposed to its own specific information. In this case, the shaded entity represents the "Building Elements" data module and the hatched entity and attributes represent part of the "Specification" data module as shown in Figure 8.15.

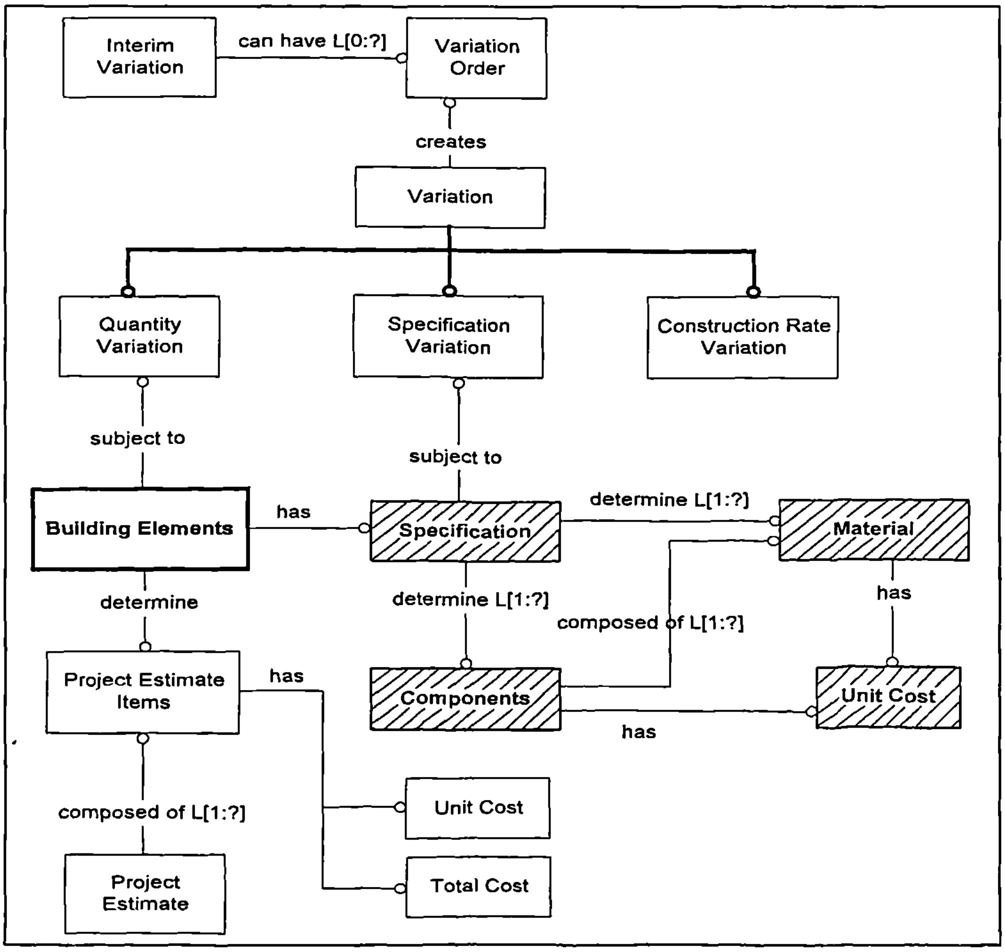


Figure 8.15: Part of EVALUATOR data model [Underwood & Alshawi, 1997]

The production of Bill of Quantity (BQ) requires the physical dimensions and the cost from the "Building Elements". This is depicted by the "Project Estimate Items" entity, which has the "Unit Cost" and "Total Cost". The "Total Cost" entity represents the total cost of the estimated item, while the "Unit Cost" entity depicts the all-in-one rate, i.e. the item's cost per quantity.

When generating monthly interim certificates, the "Quantity Variation" can be subjected to the "Building Elements" data module. This process happens during the construction of a project, for example, when the architect instructs a variation to be made to the original design. This indirectly could change the building design element's quantity, specification or addition of new elements.

(2) Relationship with SPECIFICATION

The main objective of SPECIFICATION is to generate the specification of the design element as soon as it is declared in the "Building Elements" data module [Underwood & Alshawi, 1997]. Figure 8.16 depicted the "Specification" entity which is determined by one or many "Components" and "Material". The "Components" is composed of the "Building Elements" and the "Material" where the "Material" represents the material(s) from which the components are constructed. For example, in a cavity wall, the outer leaf can be built from bricks and the inner leaf from blocks. Each "Component" and "Material" has a "Unit Cost" from which the element's total "Components" or "Material" unit cost can be built.

New four attributes have been added to the "Building Elements" to cater for specific cases, as shown in Figure 8.16 in the hatched attributes. The "Elements Specification Reference" is used to reference each building element to its specification. The "Reinforcement Length" and "Links Length" are used for storing the length of the reinforcement and links respectively. These can be referenced to the structural application in future developments. For instance, when a reinforced concrete column is created, its reinforcement length and links length must be determined. The "VR Texture Reference" attributes is added to specify the texture reference of the element displayed in VR.

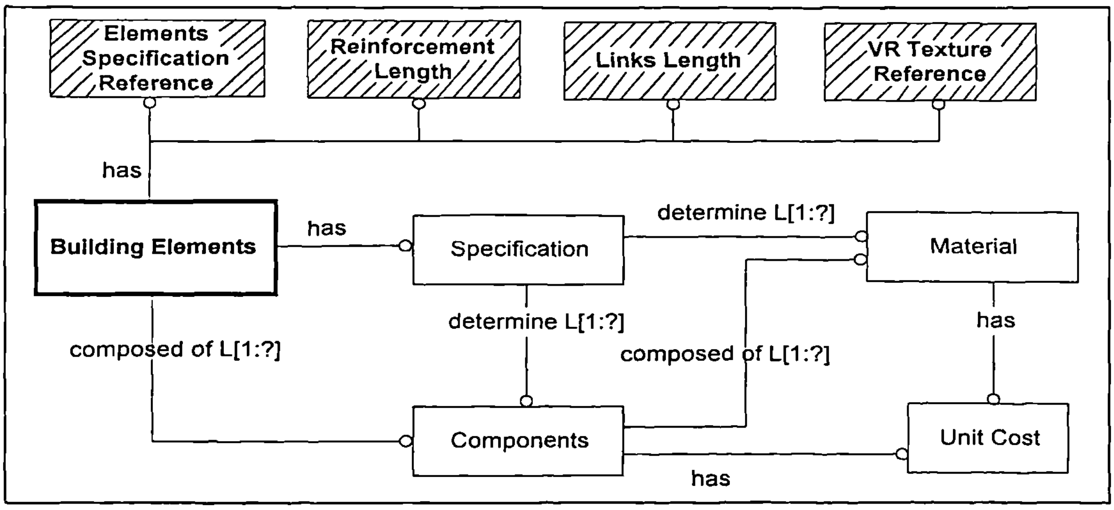


Figure 8.16: Part of SPECIFICATION data model [Underwood & Alshawi, 1997]

(3) Relationship with CONPLAN

The main objectives of CONPLAN are to dynamically generate project specific construction plans, and evaluating the buildability of the design based on the generated construction activities [Hassan, 1997]. Figure 8.17 illustrates four main entities of CONPLAN, i.e. the "Construction Space", "Project Construction Plans" and "Construction Activities". The shaded entities represent the "Building Elements" data module whereby the hatched entity represents the "Specification" data module.

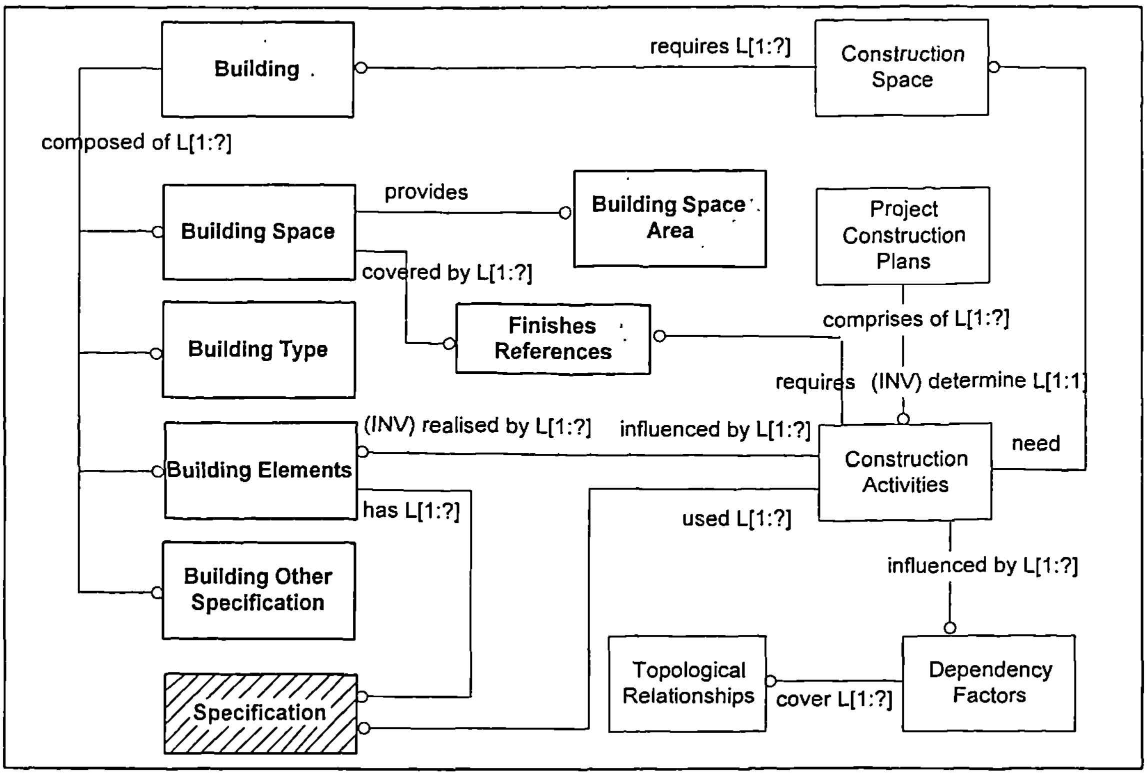


Figure 8.17: Part of CONPLAN data model [Hassan, 1997]

The generations of elemental construction activities are highly dependent on the "Building Elements" data module and their "Specification". Figure 8.17 depicts the relationship between the "Building Elements" data module and the "Specification" data module with the "Construction Activities" entity. The "Construction Activities" are influenced by one or many "Building Elements" and use one or many "Specifications". Inversely, the "Building Elements" are realised by one or many "Construction Activities".

The "Construction Activities" need "Construction Space" for executing all the construction activities at the construction site. As such, "Construction Space" requires information about the "Building" to be constructed which include the "Building

Space". The "Building Space" provides information on "Building Space Area" and the "Floor Space" as discussed earlier in section 8.6.1. The "Building" is also composed of one or many "Building Other Specification" such as "Zone" which provides the "Construction Space" working areas.

The relationship between "Construction Activities" are influenced by one or many "Dependency Factors". The "Dependency Factors" entity constitute the most important relationship between CONPLAN and the "Building Elements" data module. It covers the physical conditions, i.e. the "Topological Relationships". As discussed earlier in section 8.6.3, the "Building Elements" entity has zero or many "Topological Relationships" attributes which might be "embedded in", "supported by" or "attached to". This information is important for controlling the sequence of the "Construction Activities".

The other important information which is provided by the "Building Elements" data module is the "Elements Quantity" which is calculated from the physical dimensions of the element. This information is used by the CONPLAN to facilitate the calculation of the duration for each activity. On the other hand, the "Room Space" contains in the "Building Space" entity which is covered by finishes references such as "Floor Finish Reference", "Ceiling Finish Reference" and "Wall Finish Reference" as shown earlier in Figure 8.9. These references provide CONPLAN with the type of finishes for each space in order to set up the appropriate activities for those finishes.

(4) Relationship with INTESITE

One of the main objectives of INTESITE is to dynamically produce site specific layout information for a given design [Sulaiman, 1997]. INTESITE is totally dependent on the construction plan, i.e. CONPLAN provides most of the input such as construction activities, construction resources, construction method, etc. The relationship with the "Building Elements" data module is only bounded with the "Building" and the "Building Space", i.e. the "Building Elements" which provides the spaces in the building.

Figure 8.18 depicted part of INTESITE data model, which includes "Site Information", "Site Layout Plan", "Site Space" and "Temporary Facilities". The shaded entities represent the "Building Elements" data module and the hatched entity represents "Specification" data module. "Site Layout Plan" is influenced by one or many "Site Information" which comprises of the "Building Location", the "Site Boundary", etc. The "Building Location" or its co-ordinates provided by the "Building" entity is important in determining the space available for construction and materials/plant storage.

The "Site Space" is one of the main considerations for the site layout plan. "Site Space" is required by one or many "Temporary Facilities". Some "Temporary Facilities" can be located inside the "Building". For example, completed rooms can be used as a site office and bricks can be located on the completed floor-slab. Therefore, the information about the "Building Space" including the "Floor Space" and "Room Space" are important attributes which are provided by the "Building Elements" data module.

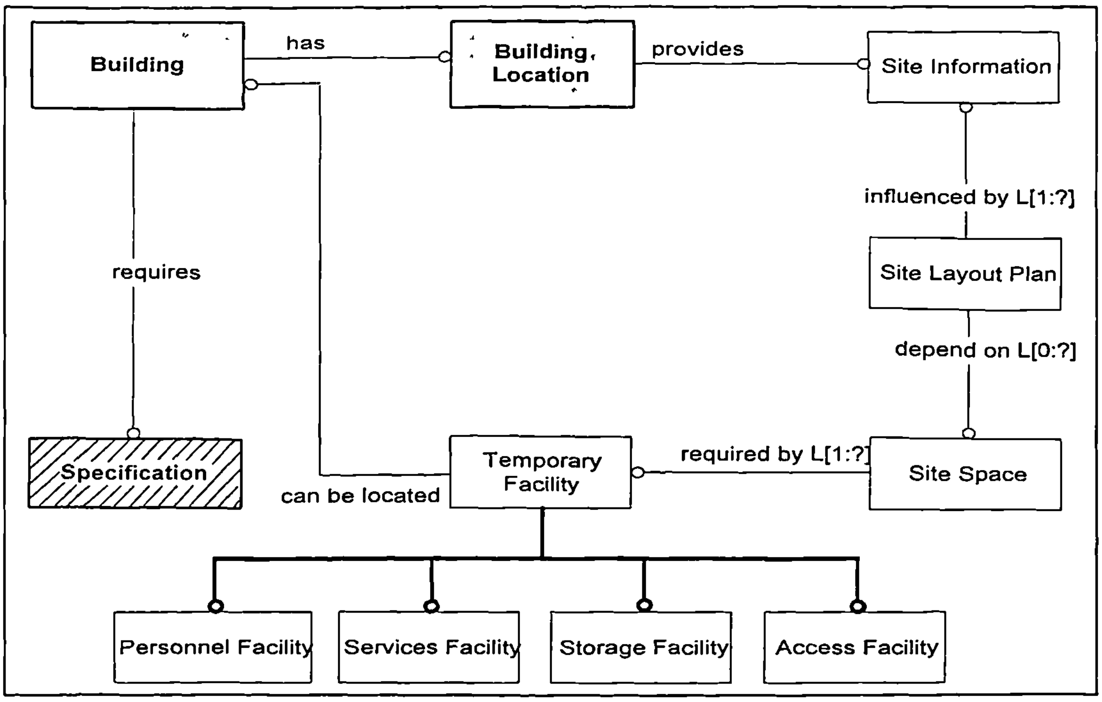


Figure 8.18: Part of INTESITE data model [Sulaiman, 1997]

(5) Relationship with CONVERT

One of the main objectives of CONVERT is to dynamically generate a virtual environment that reflects a particular construction application view of the data [Alshawi & Faraj, 1995]. CONVERT is not a construction application but it is needed to support the applications, which perform functions within the project life cycle by mapping the views of these applications to the virtual environment. CONVERT generated virtual reality models for the design elements created by CAPE at real time.

Figure 8.19 shows an overview of CONVERT prototype environment where the data flows to CONVERT, i.e. from the graphical file (DXF file) to the project model.

The data in the project model is populated by the design application (AutoCAD-AEC^{TM}). While the DXF files mirror the instances created in the "Building Elements" data module. For example, a solid wall with a brick texture in the "Building Elements" is displayed in the virtual environment as a solid wall, which has a specification of "Common Brick Type A".

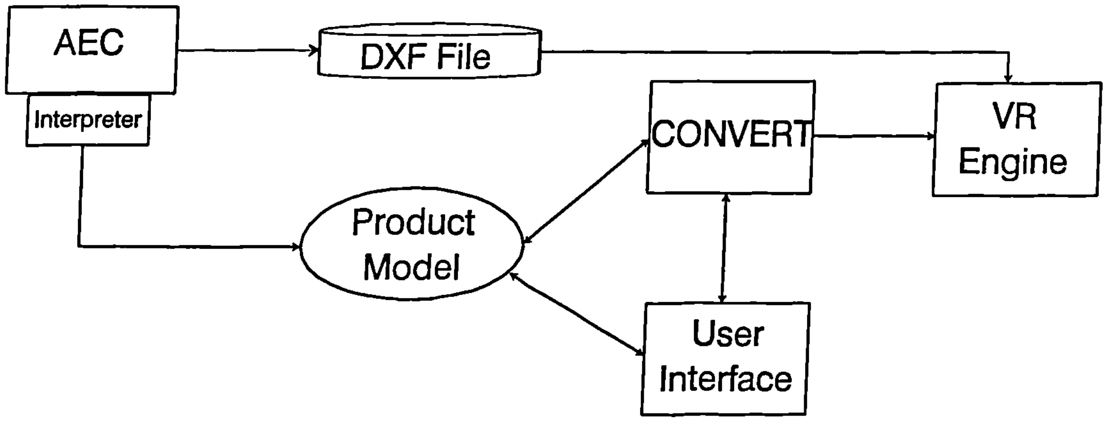


Figure 8.19: An overview of CONVERT prototype environment [Alshawi & Faraj, 1995]

8.8 Summary

The data models, which have been developed for the building elements data module as part of the ICE, are presented in this chapter. The data models have been decomposed at various levels of abstractions, beginning with the context diagram, which portrays the framework for the presentation of the ICE. The Level 1 diagrams illustrate the main entities, which form part of the project including the project specific information and the project type. The Level 2 diagrams depict more details of the building including all the specific information of the building such as building space, building design, building elements, and building other specification. The relationship between the building elements data module with other modules has been discussed.

These data models provide the foundation for the implementation process. The constructed data models can be transformed into objects, attributes, etc. and then incorporated into an object-oriented environment. The implementation into the object-oriented environment will be discussed in the following chapter.

Chapter 9

CAPE: System's Development

9.1 Introduction

The conceptual model of the "Building Elements" data module developed in Chapter 8 has been mapped into the object-oriented knowledge-based environment. Such environment is normally triggered off by feeding in project specific information through a design package, for example, a CAD system where large amount of project information can be extracted [Dym & Levitt, 1991]. This chapter describes the development of the CAPE (Construction Application Protocols for data transfEr) system including the system architecture, system implementation and system application. The system architecture describes an overview of CAPE and their main components to support the integrated environment. The system implementation describes each of the process involved in the main components and how they are implemented. The system application, on the other hand, outlines how the implemented system of CAPE can be applied to serve other construction applications or other modules in the SPACE environment.

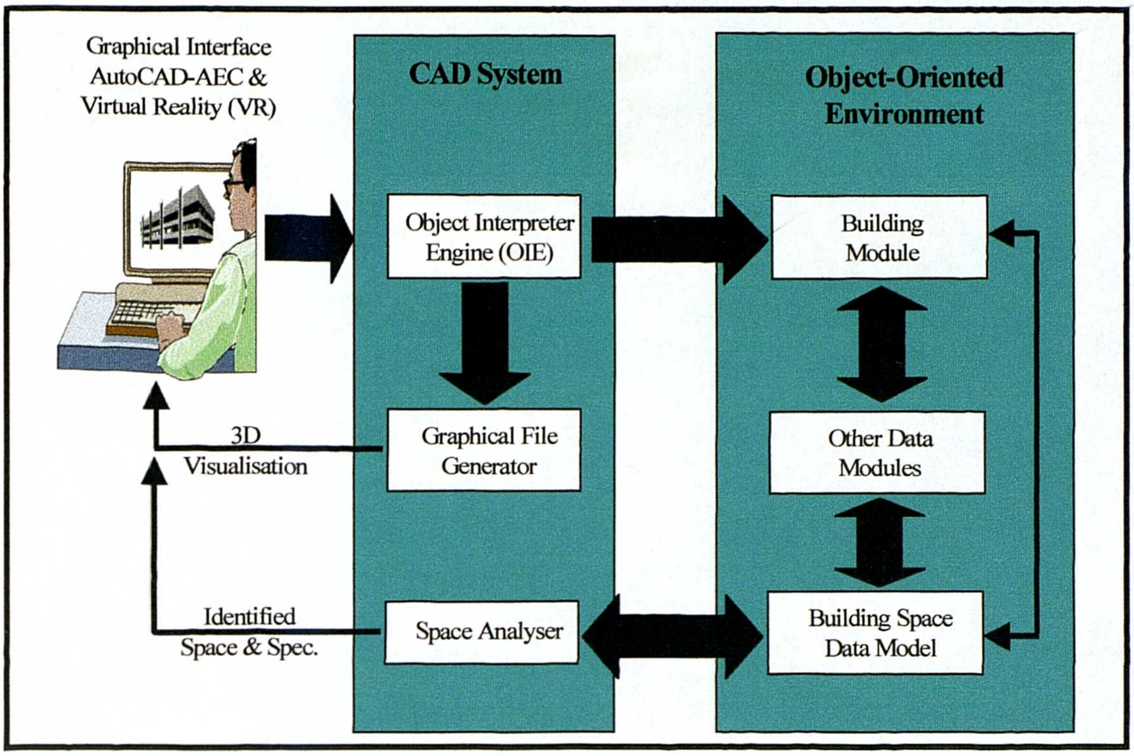


Figure 9.1: CAPE system's architecture

9.2 System's architecture

The primary aim of the CAPE module is to facilitate the flow of information from CAD packages to an object-oriented environment where it can be accessed, in a structured manner, by other applications such as construction planning, estimating, site-layout planning and VR. Therefore, in order such aims to be achieved, the CAPE system architecture has been divided into three main parts; the Graphical Interface for the input, the CAD System and the Object-Oriented Environment. Each of them provides the various important functions for the development of CAPE. The schematic diagram of CAPE system architecture is shown in Figure 9.1.

The Graphical Interface acts as an input, which allows the user to interact with the CAPE system. It consists of the AutoCAD-AEC™ graphical package and a virtual reality package (VR) for visualising the design elements in 3D. World Tool Kits™ (WTK) is used for virtual presentation of the design objects. Both tools are used by CAPE to create and obtain design data as well as visualising the design solutions.

The graphical interpretation in the CAD system is the main part of CAPE. It is divided into three parts; the Object Interpreter Engine (OIE), the Graphical File Generator, and the Space Analyser, all of them are developed within AutoCAD-AEC™. The Object-Oriented Environment, stores the interpreted objects from the CAD System in a structured manner which can be accessed by other data modules. KAPPA-PC™ is used as an object-oriented knowledge-based environment, which provides facilities such as explanations, object browser, inference tracing, etc. It also provides a KAL interpreter language, which enables users to write and test their programs. Both the AutoCAD-AEC™ and KAPPA-PC™ runs on Microsoft® Windows™ environments, which support dynamic data exchange (DDE). DDE is a facility which allows information to be shared or communicated between programs.

9.2.1 The system input

The input required by CAPE is the geometric information, which is extracted from the design package. This information describes the project design information as drawn in the AutoCAD-AEC™ package. The AutoCAD-AEC™ is an architectural package with built-in library of functions which take the architecture design a stage further by grouping a number of lines to describe a design element such as a wall, window, door, etc. The geometric information is extracted from the AutoCAD database using AutoLisp™.

9.2.2 The knowledge base of CAPE

The knowledge-based of CAPE is divided into two parts; the knowledge base which have been written within the design package and the knowledge base in the "Building Elements" data module, i.e. in the object-oriented environment.

a) The knowledge base in the CAD system

In AutoCAD, the only view of the data provided for the system integrator is the 'entity view', which shows the raw CAD data as an unordered set of graphical primitives such as points, lines and arcs [Ewen & Alshawi, 1993]. The AutoCAD-AEC™ provides the functionality for defining the design elements such as wall, window, door, etc. As a 'wire-frame' CAD-based system, a number of graphical primitive entities such as lines are connected and grouped together to form an element. Once an element is drawn, its identity as a design element is lost. It will be stored in the CAD database. Therefore, the CAD database can be considered as the prime source of building information, where if it can be extracted, it could provide potential benefits for data sharing with other applications.

In order to recapture the information about the design elements in the CAD database, special intelligent routines have to be developed. The AutoCAD-AEC™ normal routines have been modified to act more intelligent to provide such facility, which is also known as "intelligent drawing". An "intelligent drawing" refers to the use and manipulation of the CAD database in order to add substance or meaning to the entities in a drawing [McIntyre *et al.*, 1994].

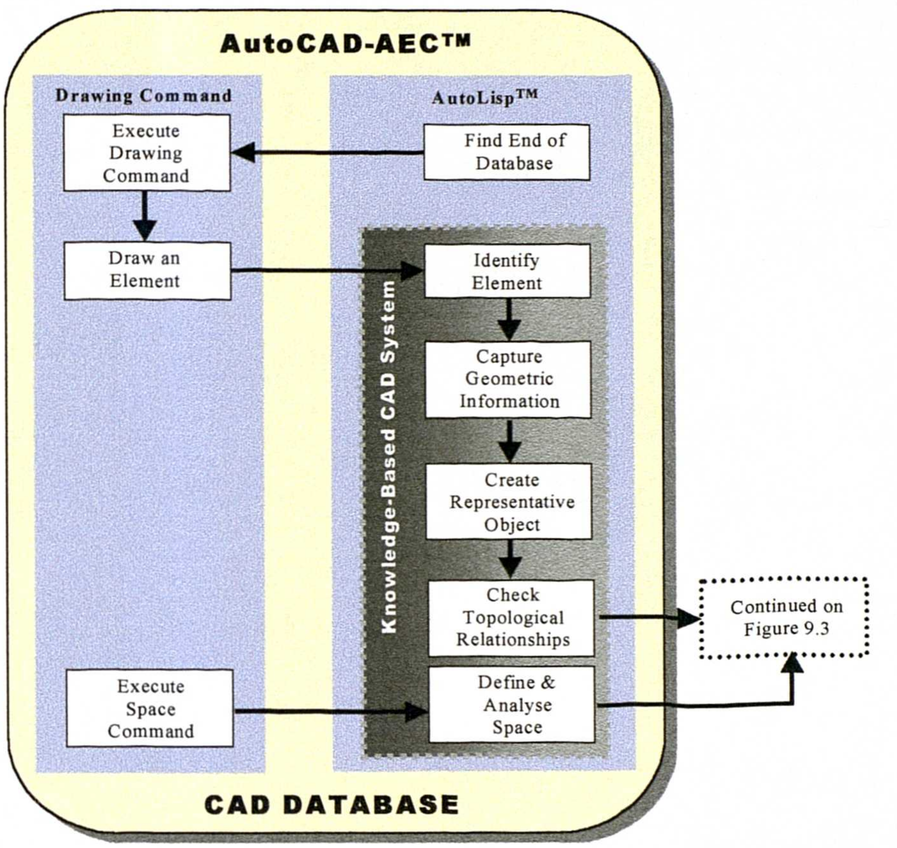


Figure 9.2: General overview of the processes in the CAD system

Several functions have been developed in the CAD system to support the development of an "intelligent drawing". These include; Identify Element, Capture Geometric Information, Create Representative Object and Check Topological Relationships, all of which form the Object Interpreter Engine (OIE). Such functions have been performed through the use of AutoLisp™ programming language. Other developed functions in the CAD system is the "Define and Analyse Space" which is used for identifying spaces of a building plan whereby the information such as space boundaries, space separator, etc. can be captured. Figure 9.2 depicted the general functions in the CAD system. Each of these functions will be discussed further in the system implementation.

b) The knowledge base in the object-oriented environment

Once an element has been created and captured in the CAD system, an object will be created in the object-oriented knowledge-based system and attached itself to the right class in the elemental hierarchy, i.e. creating a new instance for an element in the "Building Elements" data module. The created object will carry the identity, the geometry and the topological relationships of the object.

In order for such tasks to be performed, several knowledge-based functions have been developed in the KAPPA-PC™. These functions include; (a) Classify Information for classifying the incoming object into either a design element or a space; (b) Create Space Object, Define Space Object, Defined Space Associated Elements and Define Space Boundaries for Space Analysis, and (c) Create Object and Attach Information, Define Specification, Define Topological Relationship, and Check Beam Type (if an object is identify as a beam) for the Object Analysis. Figure 9.3 depicted the general functions in the object-oriented environment. Each of these functions will be discussed further in the system implementation.

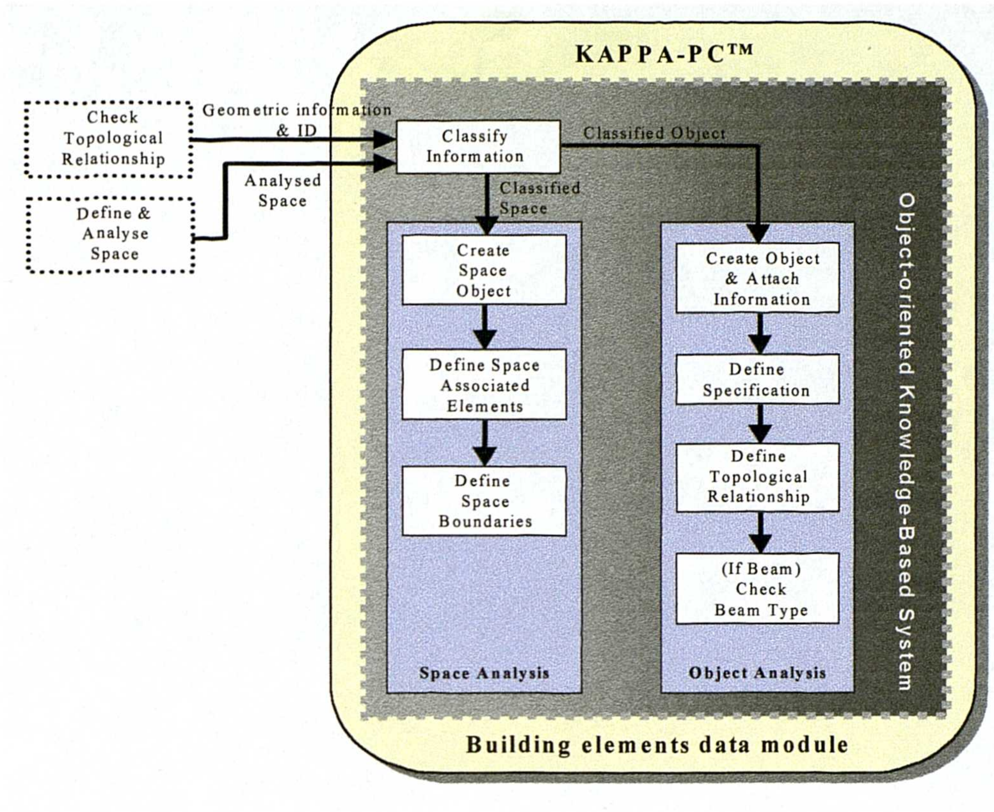


Figure 9.3: General overview of the processes in the object-oriented environment

The knowledge-based in the object-oriented environment has been developed for the following reasons:-

1. The representative object created in the CAD system and their relative information need to be stored into their right class to facilitate the exchange of information with the other data modules. Such a knowledge base should be well structured in order to overcome data sharing and integration problems.

2. The material's specification need to be defined for the created object which are required by the estimating and planning modules.

3. The topological relationship, i.e. associated with relationship, which is initiated in the CAD system need to be further expanded. For example, other relationship such as “supported-by”, “attached-to”, etc. need to be determined for other applications.

4. In a case where a beam is created, its type has to be checked. When a beam is created in line with the previous beam(s) and is connected to them, it needs to be differentiated as either a “simple-supported” or a “continuous”.

5. In a case where a space is defined, a space object need to be created and attached to its right class. For instance, when a “living room” is defined in the CAD system, an object, which represents the defined space, is instantiated at the “living room” class. This is to ensure that all the properties or information which are related to the “living room” are inherited by the created object.

6. Other knowledge such as defining the space separator and space boundaries also need to be acquired. Space separator will define the elements, which separate a space, i.e. walls, floor/slab, ceiling or roof. Space boundaries will define the co-ordinations of a space whereby a total space area can be calculated.

9.2.3 The graphical interpretation of the system's information

One of the main aims of the CAPE system is to support the flow of information from CAD drawing primitives into objects. The information held in CAPE can be visualised or viewed with the support of other graphical tools such as VR and CAD.

For example:-

1. Design objects with their relevant information in the Building Element data module can be portrayed in 3D using World Tool Kit™ (WTK). An example is shown in Figure 9.4.

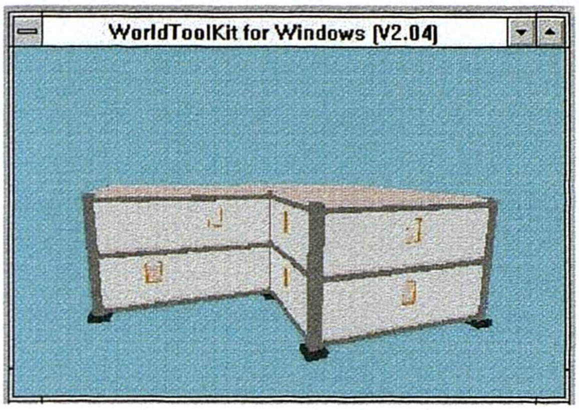


Figure 9.4: Building elements shown in VR

2. The topological relationships for each object can be triggered off in VR, for example, deleting an object such as the foundation. The relevant topological relationships with other objects are then displayed. An example is shown in Figure 9.5 (The elements affected by the deletion of the foundation are shown in blue).

Figure 9.5: Topological relationships between a pad foundation and other structural elements

3. The determined space information such as the space boundaries and the specifications of the floor surface finishes can be displayed in the CAD system by highlighting the space boundaries and pop-in the floor surface finishes. An example is shown in Figure 9.6.

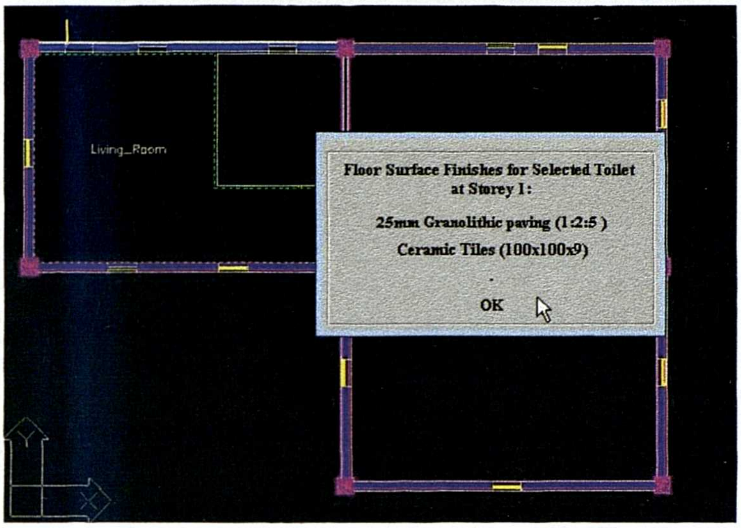


Figure 9.6: Space boundaries and floor surface finishes

Details of the system output with the demonstrations will be represented further in Chapter 10.

9.3 The system's implementation

The implementation of CAPE is based on two main parts, i.e. the CAD system and the object-oriented knowledge-based system as shown in the Figure 9.1. Each part has its own main applications, which are needed by CAPE to achieve its objectives (as described in Chapter 1). Such implementation has been developed as follows:-

9.3.1 The CAD system

In the CAD system, three main applications have been developed whereby each of them has been especially written to add more functionality to the CAD system. Such applications are the Object Interpreter Engine (OIE), Graphical File Generator and Space Analyser as shown in Figure 9.1 of CAPE system architecture.

9.3.1.1 Object Interpreter Engine (OIE)

The main objective of this application is to generate the element's representative objects. This application is especially written to transfer the "wire-frame" CAD packages into object-oriented environments where each design element is represented by an object. As mentioned earlier, the recognition of the design elements is important since the information in the CAD systems are normally in the form of lines, arc, etc.

Due to the different architccture of the available CAD systems, the interpretation and the recognition of design elements will be different. It could be for example either by layer or by pre-defined block. A layer is similar to the transparent overlays used in many traditional drafting applications. Whereas, a block is a group of entities (lines, arc, circle, point or text) [Jones & Martin, 1991].

In order to implement the OIE, several steps have been taken into considerations. Such steps are shown in Figure 9.7 and the following gives a detailed explanation of each steps:-

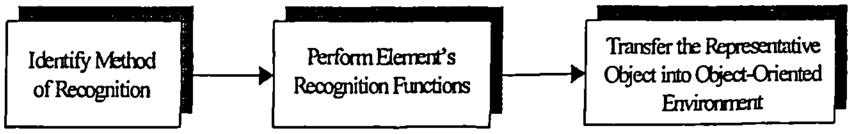


Figure 9.7: Steps for the implementation of OIE

i) Identify method of recognition

In developing "intelligent drawing" within a CAD system, researchers have used several methods of recognition. For example, McIntyre *et al.* [1994] used a specific lines (or other entities) in the drawing with a series of attributes associated with them to describe a particular entity in the drawing. Other researchers such Ewen and Alshawi [1993], used a 'layer' name in the AutoCAD-AEC™ to recognise the elements. In this study, the method of recognition using the 'layering' technique has been used and further discussion will be based on the 'layering' technique used in the AutoCAD-AEC™.

In order to allow for better use of AutoCAD-AEC™ in construction and other industries, AutoCAD-AEC™ as from Version 2.0, employs a new layering convention which is based on the European Construction Index, CI/SfB code (Samarbetskommittén för Byggnadsfrågor) [CI/SfB, 1976]. The SfB code was originally developed in Sweden and has become the standard method of filing documents in architects and other practitioner's offices. The layering convention pre-supposes some knowledge of the two coding methods recommended in BS 1192 Part 5, both of which have been used in the construction industry for many years. The reason for such a coding system, which is based on standard SfB code, is for compatibility, where information is shared between different systems.

According to AutoDesk Ltd. and AutoCAD User Group (AUG) [AutoDesk, 1993], there are at least four reasons for a layer naming convention, i.e.:-

- to rationalise information transfer
- to create a common user environment
- to give users guidance in structuring their drawing file
- to provide a structure for quality control over users' drawings

The layer naming system which is used should be suitable for, (i) any application in the building industry, (ii) any building operation: new building and rehabilitation work, (iii) manual and software control operation, (iv) drafting and modelling CAD, and (v) any CAD software [AutoDesk, 1993]. Table 9.1 illustrates the structure of AutoCAD-AEC™ layer naming convention used in this study.

Field	1	2	3	4	5	6	7	8
expressing	DISCIPLINE	CATEGORY	GRAPHICS	GRADE	LEVEL	STATUS	SCALE	TIME
Character	C	AAA	A	N	AA	C	C	A
e.g.	A	210	H	1	01	N	D	1
meaning	architecture	ext. wall	hatching	thin pen	level one	new	detail	phase one

C = Character only   N = Number only   A = Alphanumeric (Character or Number)

Table 9.1: AutoCAD-AEC™ layer naming convention [AutoDesk, 1993]

The first 6 digits (Field 1 - 4) of a layer are related to BS 1192 Part 5, the next 2 digits are used for the level starting from 00 to 99 (Field 5), whilst the remaining 3 digits (Field 6 - 8) are available to the user, if required. For example, in Field 6, it is used for status, i.e. either new [N], existing [X], or to be removed [R]. Whereas Field 7 is for scale, indicating the detail of the information being shown, and Field 8 is for time, to show the various phases of the work or to show different alternatives.

The layering convention may use several codes to identify certain elements. This makes it difficult to transfer the required information to other applications such as planning package. For example, when a cavity wall is created, four lines are drawn as a result. These lines are assigned with four different layer convention codes such as A220G700, A213G500, A212G500, and A212G700. CI/SfB code for '212' refers to the outer leaf of external walls, '213' refers to inner the leaf whilst, and '220' refers to internal walls (partitions). These four codes are used together to identify an external cavity wall.

In this study, the principle of capturing the element using layer convention and representative object (line) is based on previous work done by Ewen and Alshawi at the University of Salford [Ewen & Alshawi, 1993]. For example, when a cavity wall is created, a representative object will be created at the centre of the cavity wall by a line. This line will be assigned with a new layer such as 'CAPE210G1' which represents a layer for a cavity walls. SfB code '210' refers to external walls. While codes 'G' and '1' are not necessary for defining the elements because they only refer to the graphics, i.e. 2D Graphics. The description of this method is presented in the following section.

ii) Perform element's recognition functions

The drawing interface in AutoCAD-AEC™ consists of lists of pre-defined components in pop-down menus. Although it is not possible to gain control of the compiled code for these drawing commands, it is possible to gain control of the menu lines which execute them, i.e. by executing a function before and after execute drawing command. All the functions are developed using AutoLisp™ commands.

In order to perform element's recognition functions, several processes have to be developed which are previously shown in Figure 9.2. These are:-

- Find end of database
- Identify element
- Capture geometric information
- Create representative object
- Check topological relationship

Find end of database

The main aim of this process is to set a pointer to the end of the drawing databases. This is to assist in capturing the drawn entities without mixing with other previous entities. When the drawing command is executed, the entities, which it creates, can easily be located, i.e. they lie between the pointer and the end of the database. This process must be executed before executing the drawing command.

Identify element

As stated earlier, the technique of capturing and identify the element based on layer convention is used in this study. Using AutoCAD-AEC^{TM}, several layer names are created when an entity is drawn. For example, when a cavity wall is created, layer names such as A220G700, A213G500, A212G500 and A212G700 are created. Therefore, in order to identify an element, the algorithm below is used, i.e. by differentiating the layer name.

```
Get four characters from the layer name (e.g. 220G from A220G700)
Check the characters
    If equal to "220G" or "212G" Then the element is "Cavity Wall"
    If equal to "320G" Then the element is "Solid Wall"
    If equal to "314G" Then the element is "Windows"
    If equal to "315G" Then the element is "Door"
    If equal to "232G" Then the element is "Floor"
    If equal to "160G" Then the element is "Strip"
End of checking
```

There are some problems is using AutoCAD-AEC^{TM} since it is primarily designed for architectural drawings only and does not fully cover the structural drawing. For example, when a beam or a column is drawn, one option is used, i.e. using the same drawing command. Therefore, AutoCAD-AEC^{TM} creates only one layer name. A special function has to be developed in order to differentiate between the beam and the column. The algorithm below is used for this purpose:-

Get all the y and z co-ordinates of the entity
Check the co-ordinates
    If the different of y co-ordinate is equal to 0 AND
    If the different of z co-ordinate is NOT equal to 0 Then
        The element is "Beam" Else
        The element is "Column"
End of checking

Capture geometric information

In AutoCAD-AEC^{TM}, there are two different types of entity which are used to draw an element, i.e. line and polyline (normally come from 'Block'). Thus, the method of capturing the geometric information of an element will be different. For this purpose, two special functions have been developed to group the line and polyline entities. The first function sorts the entities according to the "Line" and "Vertex" only and then group the entities into a list. The second function, on the other hand, first explodes the 'Block' entity into several polyline entities, then sorts the entities according to the "Polyline" only and groups the entities into a list.

The examples and the algorithms below will show how the geometric information of an element are captured:-

1) Example of 'Cavity Wall' entity database:-

    ((-1 . <Entity name: 60000bc8>) (0 . "LINE") (8 . "A212G500") (5 . "A07") (39 . 2600.0) (10 3790.61 18085.0 0.0) (11 7070.93 18085.0 0.0) (210 0.0 0.0 1.0))

Height of cavity wall	= 2600.0 mm [from (39 . **2600.0**)]
First x co-ordinate	= 3790.61 [from (10 **3790.61** 18085.0 0.0)]
First y co-ordinate	= 18085.0 [from (10 3790.61 **18085.0** 0.0)]
Second x co-ordinate	= 7070.93 [from (11 **7070.93** 18085.0 0.0)]
Second y co-ordinate	= 18085.0 [from (11 7070.93 **18085.0** 0.0)]

Whereby '39' represents the code for height, '10' represents the code for the first co-ordinates, and '11' represents the code for the second co-ordinates.

2) Example of a "Beam" entity database after sorting the list:-

((-1 . ,Entity name: 60000a5c>) (0 . "3DFACE") (8 . "S280G600") (5 . "15E")
(10 6850.0 11000.0 400.0) (11 6550.0 11000.0 400.0) (12 6550.0 6000.0 400.0)
(13 6850.0 6000.0 400.0) (70 . 0))

Beam width = 300mm (6850 – 6550)

↦ 6850 from (10 **6850.0** 11000.0 400.0)

↦ 6550 from (11 **6550.0** 11000.0 400.0)

Beam depth = 400mm [from (10 6550.0 11000.0 **400.0**)]

Beam span = 5000mm (11000.0 – 6000.0)

↦ 11000.0 from (10 6850.0 **11000.0** 400.0)

↦ 6000.0 from (12 6550.0 **6000.0** 400.0)

Create representative object

The principle of creating a representative object is based on Ewen and Alshawi [Ewen & Alshawi, 1993]. The main aim of this function is to group the list of entities, which represents an element into a single entity, i.e. representing a single object without loosing the relationships between the newly created object and its original entities. The object is created as a line on a hidden layer in the drawing. The representative object is cross-referenced to all its constituent entities to subsequently allow its geometry to be determined, and to allow the complete component to be identified from any one of the lines on the drawing.

The representative object is created as a central line for cavity and solid wall, door and window, as a centre point line along the beam and column, and as a diagonal line for footing and floor. An example of creating a representative object for a cavity wall is shown below.

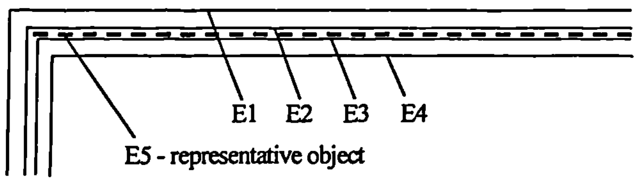


Figure 9.8: The representative object for a cavity wall [Amended from Ewen & Alshawi, 1993]

The four cavity wall entities labelled E1, E2, E3 and E4 in Figure 9.8 are the lines created by the original drawing command and the dotted line E5 is the representative object created for the cavity wall as a centre line.

Entity data			**Comment**
Layer name	Entity handle	Xdata	
A220G700	E1	E5	Wall entity
A213G500	E2	E5	Wall entity
A212G500	E3	E5	Wall entity
A212G700	E4	E5	Wall entity
CAPE210G1	E5	E1, E2, E3, E4	Representative object

Table 9.2: The cavity wall entity cross-reference scheme [Amended from Ewen & Alshawi, 1993]

Table 9.2 illustrates the two-way cross-reference system, which is created between entities, and their representative object using handles stored in the extended data (Xdata). Handle is a unique, permanent identifier, which is saved with the entity in the drawing file [Jones & Martin, 1991]. Handle is automatically and permanently assigned to an entity upon creation and remains with it unless it is deleted. Xdata is the extended entity data provided by AutoCAD™, which allows users to add any type of data to the AutoCAD™ database using AutoLisp™ or C programming language.

An example of cavity wall entities which show the original entity handle and the representative object entity handle where they are cross-reference is shown below:-

The original entity:

((-1 . <Entity name: 60000bc8>) (0 . "LINE") (8 . "A212G500") (5 . **"A07"**) (39 . 2600.0) (10 3790.61 18085.0 0.0) (11 7070.93 18085.0 0.0) (210 0.0 0.0 1.0) (-3 ("DIGROUP" (1002 . "{") (1000 . **"A17"**) (1002 . "}"))))

The representative object

((-1 . <Entity name: 60000d68>) (0 . "LINE") (8 . "CAPE210G100") (6 . "CON-TINUOUS") (62 . 1) (5 . **"A17"**) (10 4015.61 17860.0 0.0) (11 19144.8 17860.0 0.0) (210 0.0 0.0 1.0) (-3 ("DIGROUP" (1002 . "{") (1000 . **"A07"**) (1002 . "}"))))

Where DIGROUP is the group of handles within the same element.

Check topological relationship

As previously mentioned, it is important to establish the topological relationships between the building elements to enable other construction applications, such as planning, to perform their functions. Such relationships can be identified at the early stage of design and stored into each entity of the element. Since the AutoCAD-AEC™ is a 'wire-frame' based CAD system, the method of 'intersection' between the elements has been used. Although this method requires more processing time, it is found that this is the best method which can be used in 'wire-frame' based CAD systems. However, for windows and doors, AutoCAD-AEC™ has provided a special routine whereby a window or a door cannot be inserted without a wall. Thus, when a window or a door is inserted into a wall, extra entities are created and therefore used to capture the topological relationships between window/door and the wall.

The topological relationship between the elements using the 'intersection' method is shown in the following algorithm:-

```
Initialise the intersection list (For all elements as shown in Figure 9.9)
Get the elements ID
Get the properties of the elements
Check the level of the elements
For Column & Column, Beam & Column, Column & Pad Footing, Slab & Beam;
    If both element NOT in the same level, Then Check the intersection
    Else end of intersection check
For Beam & Beam, Wall & Wall;
    If both element in the same level, Then Check the intersection
    Else end of intersection check
Check the intersection;
    If the elements are intersect, store both elements in the list
End of Checking
```

Figure 9.9 shows all the elements, which are considered for identifying the topological relationships. The diagonal line, the hatch and the shaded area show the surface or area where the intersection is performed.

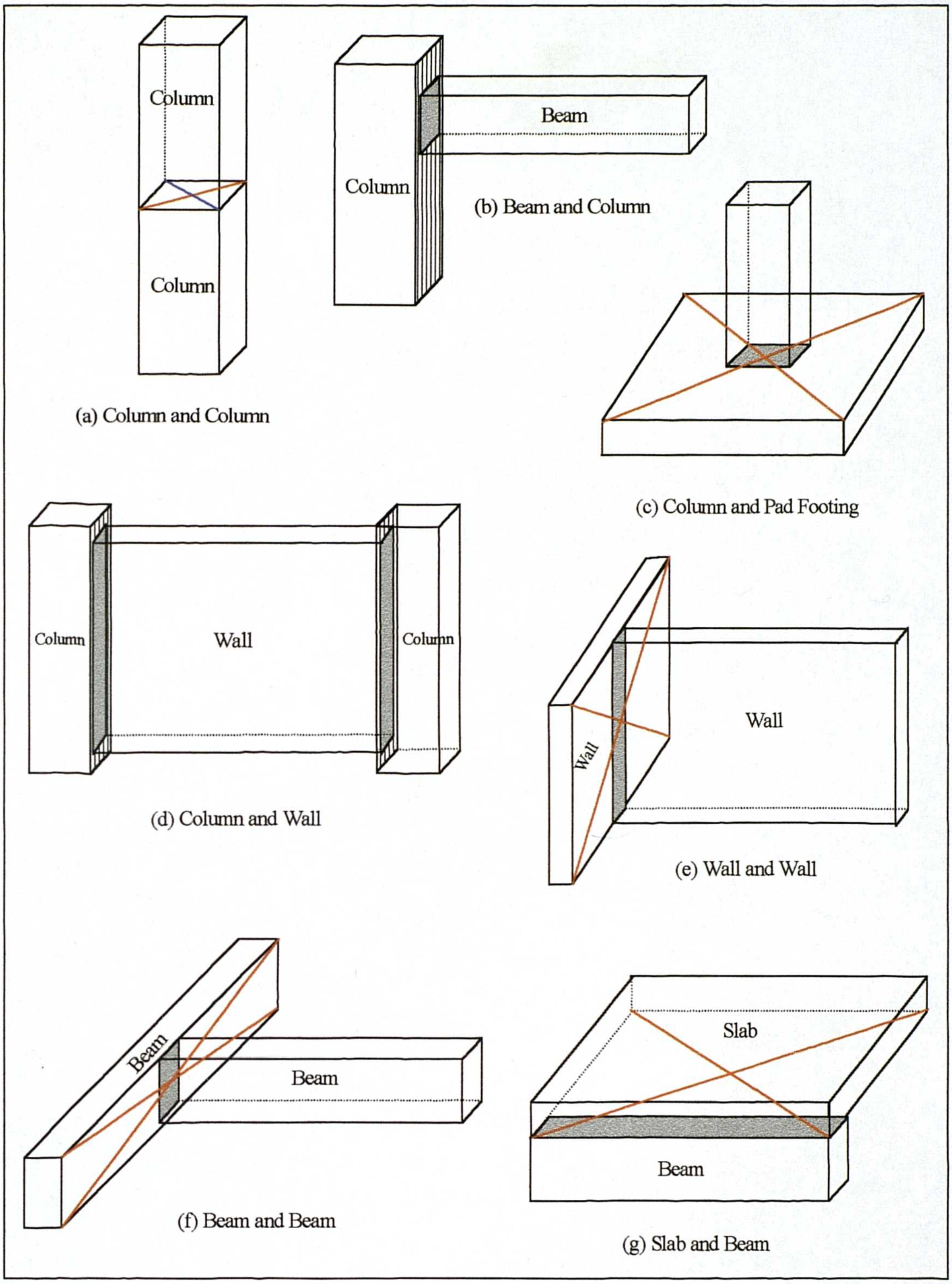


Figure 9.9: Topological relationships between elements and their intersection

The entities of the elements, which are associated with each other, will be updated to include the new entity. This is to ensure that the information of the topological relationships between the elements is kept together, thus, providing consistency between the two elements, i.e. if one element is deleted, the other element, which are associated with it will also be affected. In this study, the associated elements will be updated for the representative object only. This is to ensure that the process of maintaining and updating the representative object will be much easier and faster. An example of AutoCAD™ entity database for the associated elements is shown below:-

The representative object for beam;

((-1 . <Entity name: 2c71308>) (0 . "LINE") (5 . "4259") (100 . "AcDbEntity") (67 . 0) (8 . "CAPE284G100") (62 . 1) (6 . "CONTINUOUS") (100 . "AcDbLine") (10 3000.0 7000.0 300.0) (11 3000.0 3000.0 300.0) (210 0.0 0.0 1.0) (-3 ("DIASSOC" (1002 . "{") (1000 . "4558") (1000 . "2D4F") (1000 . "2C5A") (1002 . "}")) ("DINAME" (1002 . "{") (1000 . **"Simply_Supported_Beam"**) (1002 . "}")) ("DIGROUP" (1002 . "{") (1000 . "4251") (1000 . "4252") (1000 . "4253") (1000 . "4254") (1000 . "4255") (1000 . "4256") (1000 . "4257") (1000 . "4258") (1002 . "}"))))

The representative object of column;

((-1 . <Entity name: 2c708f8>) (0 . "LINE") (5 . "2D4F") (100 . "AcDbEntity") (67 . 0) (8 . "CAPE282G100") (62 . 1) (6 . "CONTINUOUS") (100 . "AcDbLine") (10 3000.0 7000.0 0.0) (11 3000.0 7000.0 800.0) (210 0.0 0.0 1.0) (-3 ("DIASSOC" (1002 . "{") (1000 . "4259") (1000 . "39BF") (1000 . "330D") (1002 . "}")) ("DINAME" (1002 . "{") (1000 . **"Square_Column"**) (1002 . "}")) ("DIGROUP" (1002 . "{") (1000 . "2D47") (1000 . "2D48") (1000 . "2D49") (1000 . "2D4A") (1000 . "2D4B") (1000 . "2D4C") (1000 . "2D4D") (1000 . "2D4E") (1002 . "}"))))

Where DIASSOC is the group of handles associated with other element(s).

Figure 9.10 shows the 'snap-shot' of the topological relationships. The topological relationship function is invoked as soon as an element is created.

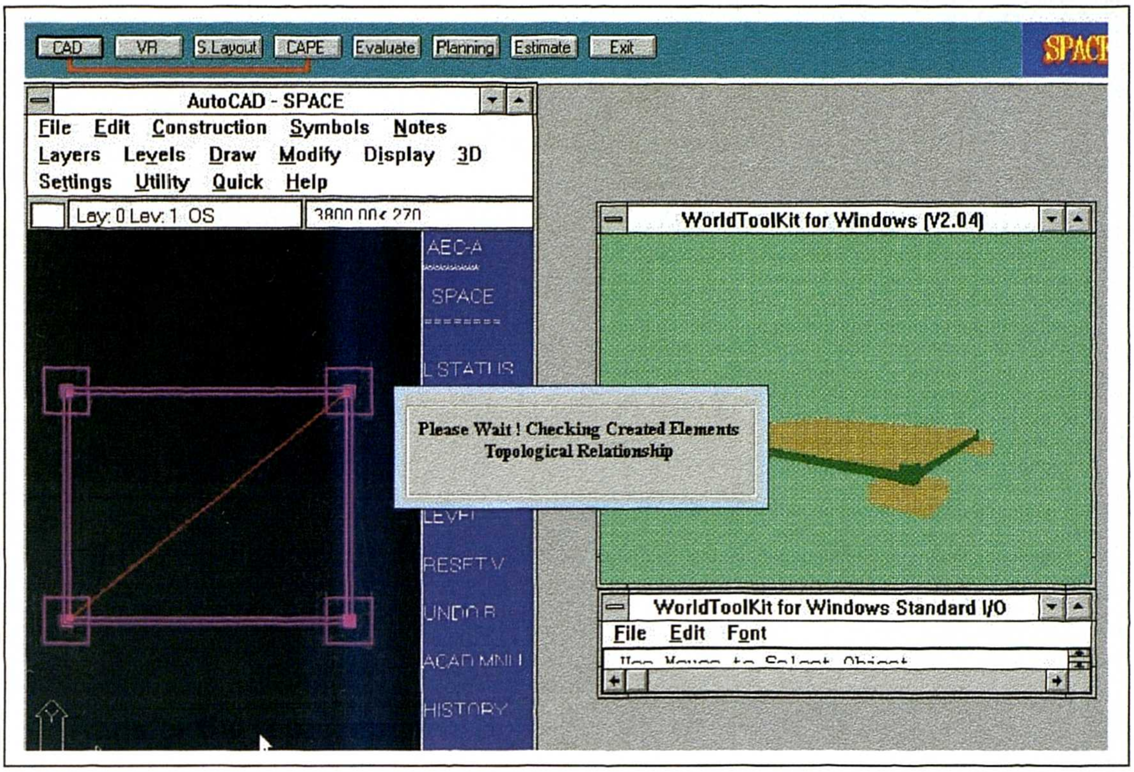


Figure 9.10: The topological relationship 'snap-shot'

iii) Transfer the representative objects into KAPPA-PC™

Once the representative objects are created in the CAD system, they are transferred into the object-oriented environment (KAPPA-PC™). A routine has been incorporated within the element recognition function to transfer the representative objects and their properties to KAPPA-PC™. The algorithm below shows how such transfer has been performed:-

Set the application name to KAPPA
Check if KAPPA and Project Model (SPACE) have already invoked
    If NOT Then load KAPPA and Project Model
Set the DDE Channel ON
Transfer the information (ID, Dimension, Handle, Locations, etc.)
If the topological exist, transfer the associated element information
Close the DDE Channel

9.3.1.2 Graphical File Generator

The main objective of this application is to generate the graphical file in DXF format which represents the representative object. This graphical file can be used by other graphical tools such as VR to visualise the objects in virtual environment or in other CAD environments. In this study, a DXF file is created for each representative object which contains graphical representation and other related information such as geometric information, topological, etc. In order to maintain the consistency between the DXF files and their representative objects in the project model, unique IDs have been used. The handle name created by AutoCAD™ has been used for this purpose. Since KAPPA-PC™ does not accept object name starting with a number, an "I_" extension has been introduced to the beginning of the representative object handle's name. This has also been applied to the DXF filenames, which started with an extension of "I_".

In order to generate a single DXF file for a representative object, all entities which are cross-referenced to the representative object are grouped together in a list. For example, a cavity wall consists of four entities and are grouped together in a DXF file.

An algorithm of generating a DXF file is shown below:-

Get the representative object
Get the representative object handle name
Get the group of the representative object (DIGROUP)
Set the group in a selection set
Create a DXF file with an extension "I_"
End of DXF file creation

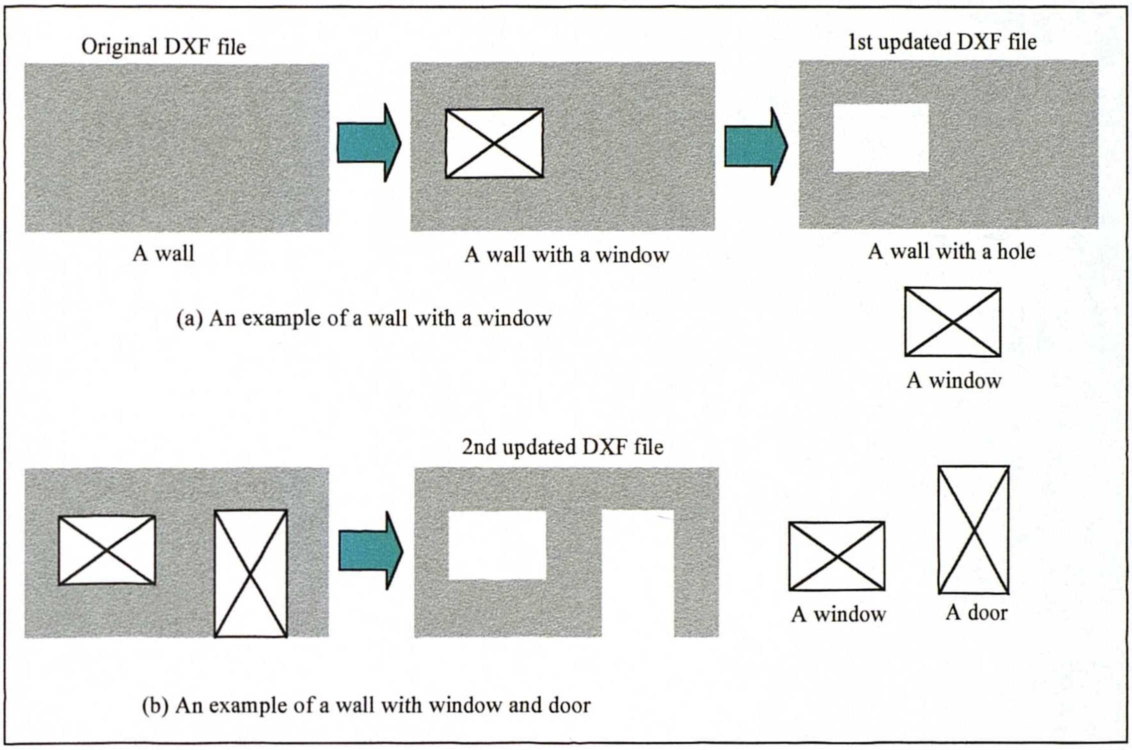


Figure 9.11: Updates DXF file for complex elements

There is a case where it is too complex to be solved by the algorithm stated above. This happens when a window or a door is inserted into a wall. Figure 9.11 shows how a DXF file is updated when a window is inserted into a wall. In Figure 9.11(a), a DXF file is first created for a wall. After a window is inserted into that wall, another DXF file is created for the window. However, the wall is viewed to have a hole when the window is removed. Therefore, the wall's DXF file has to be updated to include some new entities which creates the hole. A more complicated case can be seen when both a window and a door are inserted into a wall as shown in Figure 9.11(b). In this case, the same procedure, is applied to the wall and the window. The only additional routine is that the DXF file for the wall has to be updated twice, i.e. firstly, when the window is first inserted into the wall and secondly, when the door is inserted into the wall. The following algorithm shows how these routines are performed. This algorithm is a continuation of the previous algorithm:-

```
Get the representative object and check the ID for the representative object
If the ID is 'Window' OR 'Door' Then
    Get the new entities created for wall
    Check if the selection set for the particular wall is already exist
    If the selection set is already exist
        Then construct new selection set for the wall to include the new entities and
        Update the previous wall DXF file
        Else construct new selection set for the wall
    End of selection set check
Create DXF file for the window or door
End of DXF file creation
```

9.3.1.3 Space Analyser

The main objective of this application is to analyse the spaces which are created within the building. Although AutoCAD-AEC™ provides a function to define a space, it can only be used to define a specific space function (e.g. a living room, kitchen, etc.) and does not analyse the space to give specific information such as boundaries, elements which creates the space, etc. In this study, a special function has been developed and added to the AutoCAD-AEC™ for analysing spaces. The steps which are followed for analysing spaces, their associated elements and the space boundaries are:

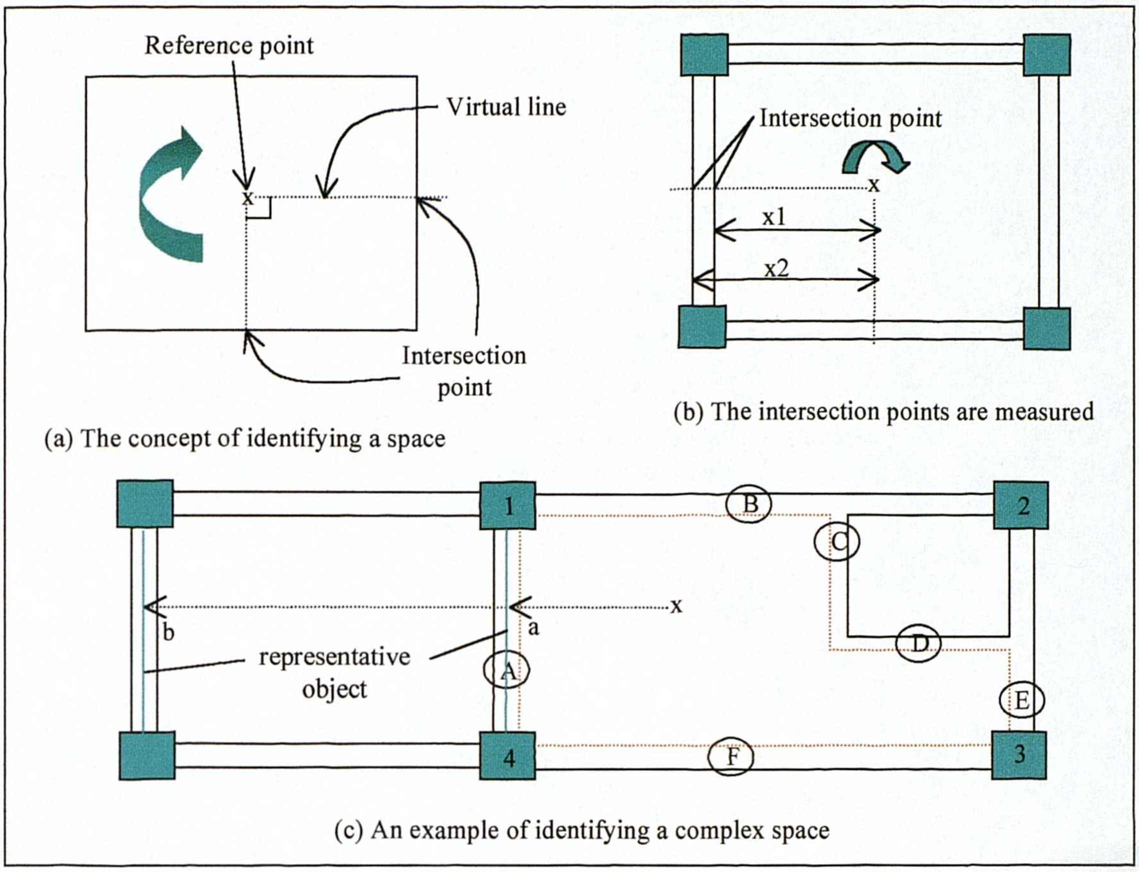


Figure 9.12: Analysing a space

1. Define a space function in a space where a space-reference point can be captured.
2. Capture all previously created walls and sort the walls according to the level the space-reference point.
3. Draw a 'virtual line' from the space-reference point to the right-hand side where an intersection point can be captured. This concept is shown in Figure 9.12(a).
4. Get the distance from the intersection point to the spaces-reference point. In Figure 9.12(b), x1 and x2 can be determined. The minimum distance, i.e. x1 will be selected.

5. The same procedure is continued using a 'virtual line' which is drawn at the top right-hand side and bottom of the space-reference point following a clockwise direction.

To illustrate the concept in more detail, consider an example of a complex space as shown in Figure 9.12(c). In this example, more than four walls have been drawn to create a space. The following steps show how the concept work.

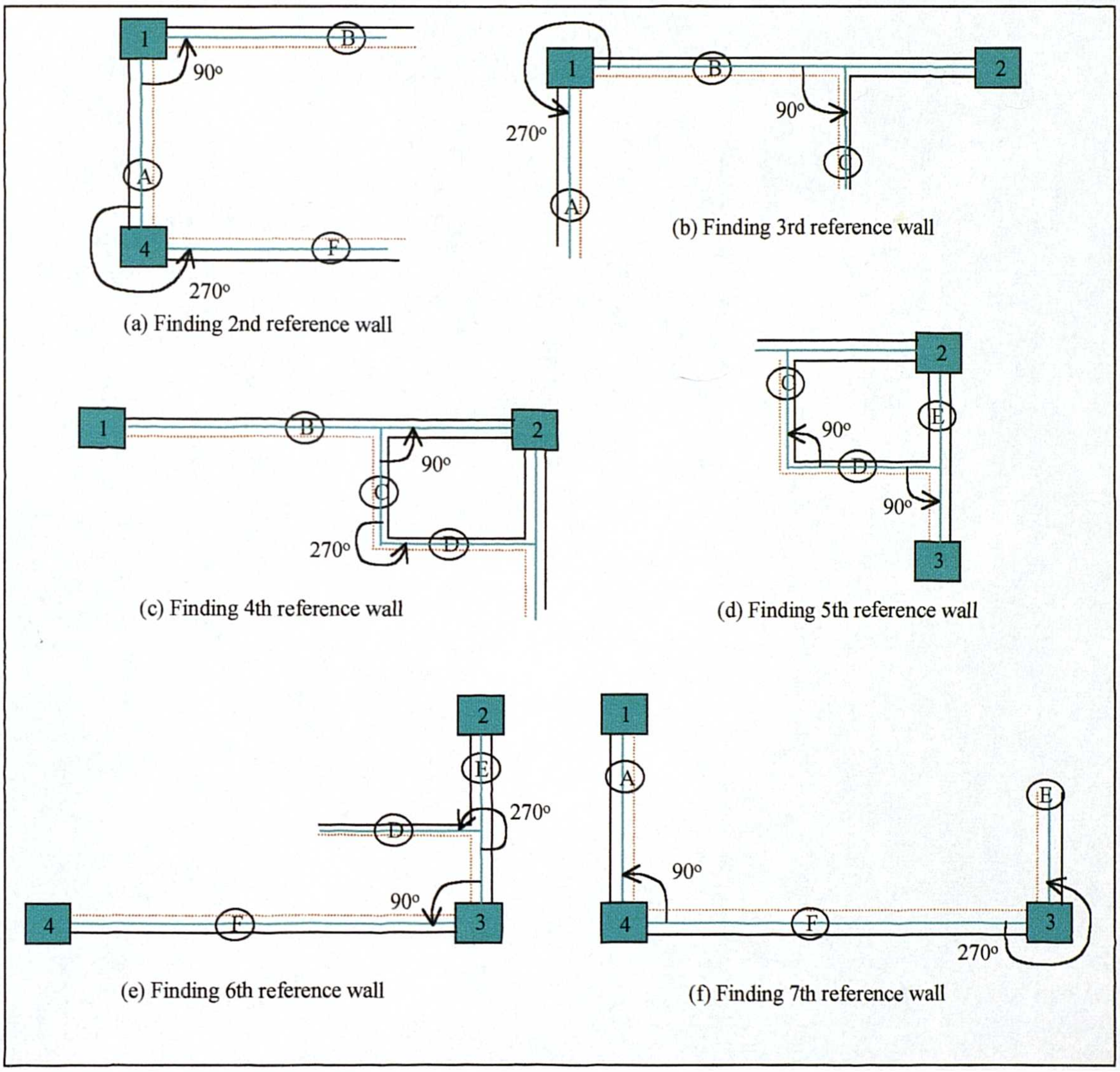


Figure 9.13: An example of identifying a complex space

1. Check all the captured walls, which are closest to the space-reference point. In this case, wall 'A' will be selected since distance 'a' is closer than distance 'b' (Figure 9.12(c)). The same procedure is done to the top of the space-reference point where wall 'B' will be selected. At the right-hand side, wall 'C' will be selected and finally at the bottom, wall 'F' will be selected.
2. Check the wall(s) associated with all the captured walls. If there is any, include the associated walls into the previous captured walls. In this case, wall 'D' and 'E' will be included since wall 'D' is associated will wall 'C' and wall 'E' is associated with wall 'D'.
3. Take the first wall from the captured walls as the reference wall. Find other walls, which are closest with the reference wall. This is done by checking the walls, which are associated with the same column associated with the reference wall and also which are included in the previously captured walls.
4. The second reference wall is identified using the angle method whereby the smallest angle will be used. This is to ensure that the space will be analysed in a clockwise direction. In Figure 9.13(a), wall 'B' will be selected since it provides an angle of 90° compare to wall 'F' which provides an angle of 270°. The measurement of an angle is automatically done by AutoCAD™ functions which measured the angle in counter-clockwise.
5. The same step 3 and 4 are followed until the final reference wall is the same with the first reference wall as shown in Figure 9.13(b) – 9.13(f). In a case where the same reference wall (which has been considered earlier) given smaller angle than the second wall, the second wall will be selected. This is shown in Figure 9.13(c) where wall 'D' is measured to have an angle of 270°, while wall 'C' has 90°. Since wall 'C' has been selected earlier, wall 'D' will be selected as the new reference

wall.

6. Finally, after all the walls have been captured to define a space, all the captured reference walls will be stored. The next step is to define the space boundaries.

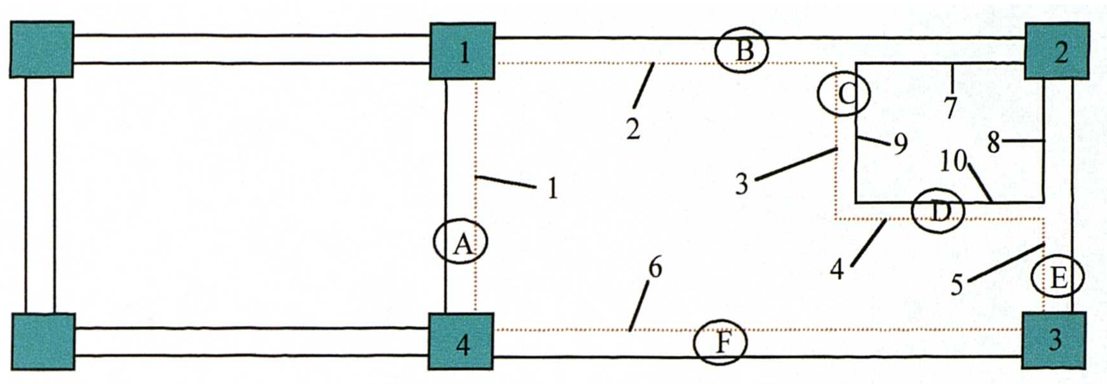


Figure 9.14: An example for analysing the space boundaries

Using the example shown in Figure 9.14, the steps involved in analysing the space boundaries can be explained as follows:-

1. For the first boundary, get the first wall from the wall reference list, i.e. wall 'A'. If the reference wall is a cavity wall, get a list of entities, which belongs to cavity wall (DIGROUP). Select the entity with the layer name "A220G7" which represents the internal wall. In this case, entity '1' will be selected.
2. If the reference wall is a solid wall, get all the entities which belong to the solid wall. In this case, two entities will be selected. The entity, which is closest to the space reference point, will be selected.
3. For the second boundary, get the second reference wall, i.e. wall 'B'. Check the wall as in the step 1 and 2. In this case, two entities will be captured, i.e. entity '2' and '7'.

4. Check which entity has their co-ordinates closest to the previous reference wall. In this case, entity '2' will be selected.
5. For the third boundary, since the third wall is a solid wall, two entities have to be compared, i.e. entity '3' and '9'. Following the same method used in step 4, entity 3 will be selected.
6. For the fourth boundary, since the fourth wall is also a solid wall, two entities have to be compared, i.e. entity '4' and '10'. Following the same method used in step 4, entity '4' will be selected.
7. For the fifth boundary, two entities have been selected, i.e. entity '5' and '8'. Following the same method in step 4, entity '5' will be selected.
8. For the sixth boundary, only one entity has been selected, i.e. entity '6'.
9. Store the selected entities which represent the space boundaries of a defined space as dotted lines (Figure 9.14).

The above steps for analysing a space, which includes analysing the space boundaries and space separator (wall, slab, etc.), provide information such as:-

- Space boundaries in which the surface finish area can be determined.
- Space boundaries in which the total space area can be determined.
- Space boundaries will define the relationship between spaces, i.e. adjacent to or far from.
- Space separator (e.g. walls) which have the associated elements (e.g. windows/doors) will provide the information for determining the total heat lost.
- Space separator which is associated with many spaces.

The following steps will show how the information provided above can be utilised using an example in Figure 9.15.

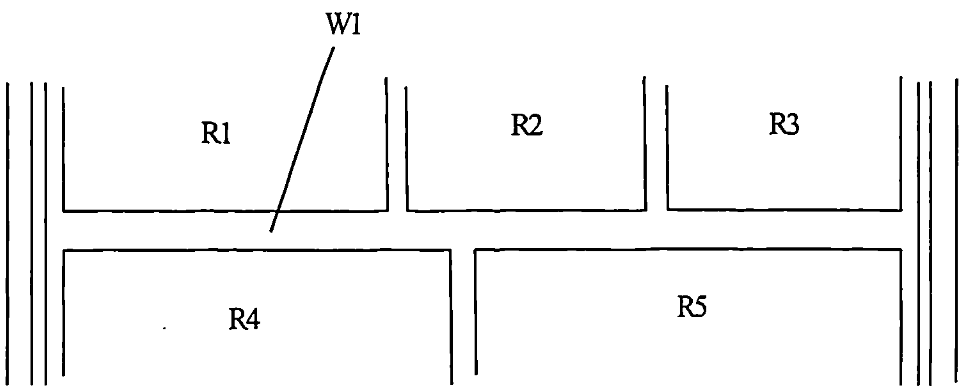


Figure 9.15: The relationship between space

1. Using the space boundary of space R4, the adjacent spaces such as R1 and R2 can be determined by finding the closest/nearest distance between the space boundary of space R4.
2. The number of spaces separated by wall W1 (R1, R2, R3, R4 and R5) can be detérmined by sorting the associated elements and spaces in the wall W1 representative object (DIASSOC).

9.3.2 The object-oriented environment

In the object-oriented environment, the building data module and other data modules are implemented. The building data module includes building elements data module, building space data module, building design data module, etc. which have been mapped from the conceptual model as described in Chapter 8. Figure 9.16 shows the object hierarchy, which has been mapped in KAPPA-PC^{TM}. Figure 9.16(a) shows the object hierarchy of the project model (SPACE) including the building module and other modules such as construction and site information. Figure 9.16(b) shows the object hierarchy of the building module including building elements data module, building space module, building design module, etc.

Figure 9.3 shows the general processes which are needed in the object-oriented environment. Two types of analysis have to be performed, i.e. the object analysis and the space analysis.

9.3.2.1 Object analysis

The main objective of this analysis is to identify the type of objects, which are created in the CAD system, create related instances, and ensure that the right information are attached to the objects such as material specifications and topological relationships. The processes are explained as follows:-

i. Create object and attach information
ii. Define specification
iii. Define topological relationship
iv. Check beam type

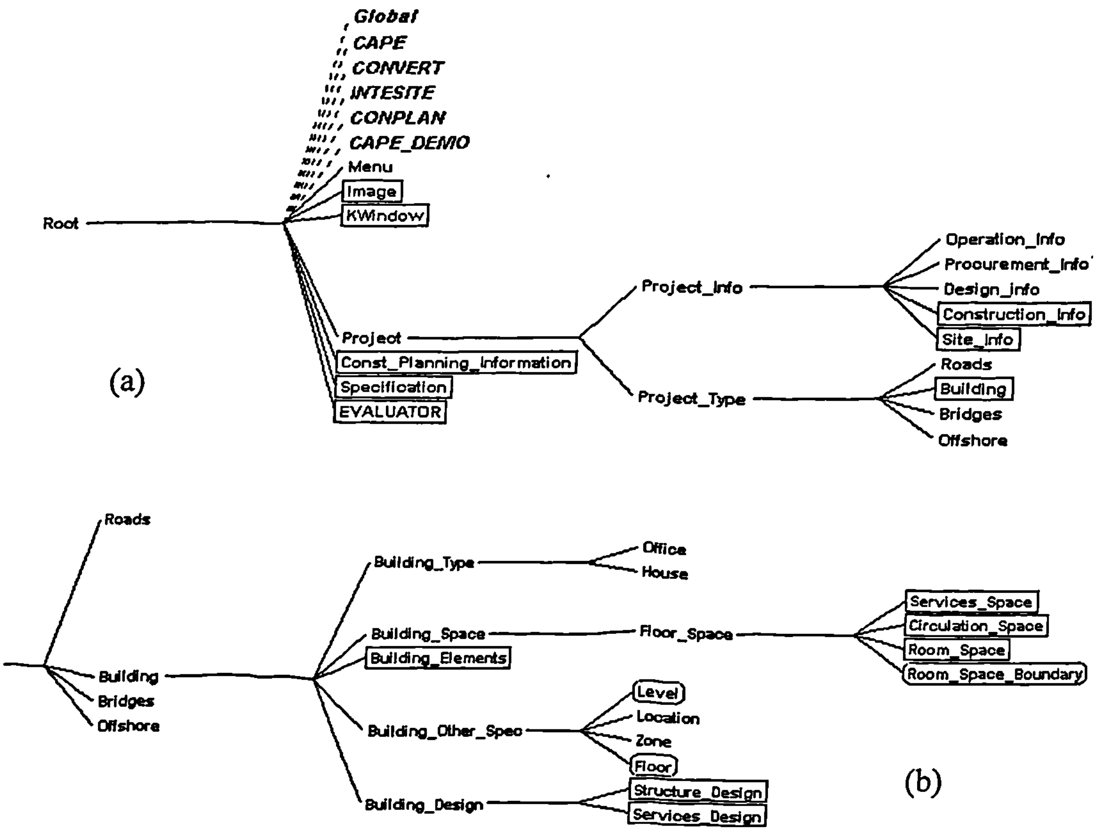


Figure 9.16: Object hierarchy in KAPPA-PC™

i) Create object and attach information

After checking the associations between the elements in the CAD systems, the ID and the geometric information of the element is transferred to the object-oriented environment. In order to ensure consistency between the new objects and their classes in the object hierarchy, the ID of the new created elements are classified according to their class type. Instances are then created in the right class in the building elements data module. For example, when a representative object is sent to KAPPA and their ID has been classified as a cavity wall, an instance is created at the cavity wall object class using the given ID as its name. As mentioned earlier, the handle name which has been provided by the CAD systems, is used for the ID. All the geometric information are then attached to that instance. Figure 9.17 below shows an example of a cavity wall instance with the slots to store the information.

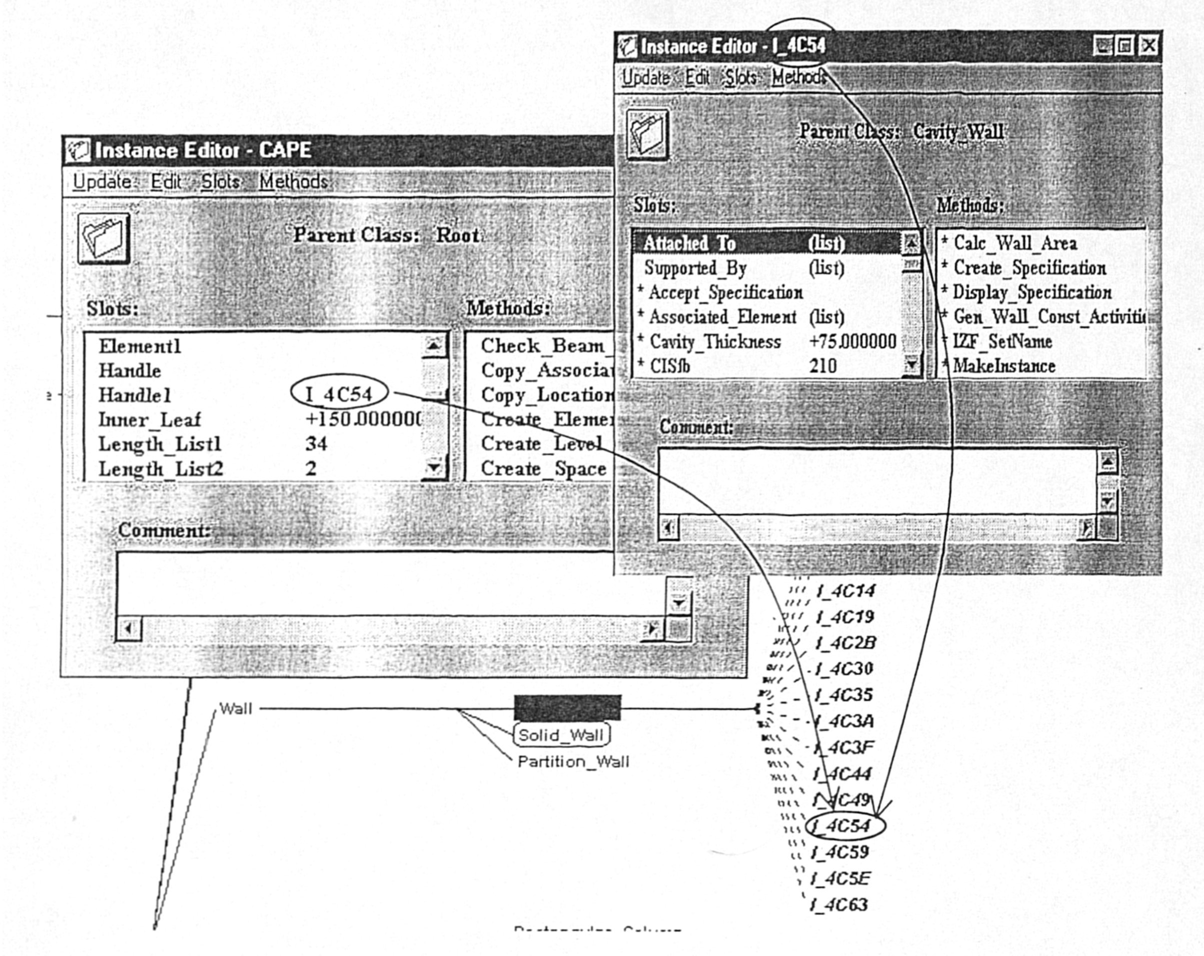


Figure 9.17: An example of cavity wall instance with their slots and values

ii) Define specification

Once the object has been created, the material specification for that object has to be defined. Firstly, previous specifications for similar objects are checked. If similar objects exist in the data module, it will automatically be copied and cross-referenced with the previously defined specifications. Otherwise, the specification data module will be triggered off and a new list of material specifications will be displayed for the input. A new specification instance is then created which will be cross-referenced to the new object. Figure 9.18 below shows a cavity wall instance, which is cross-referenced with the specification instance at the specification module.

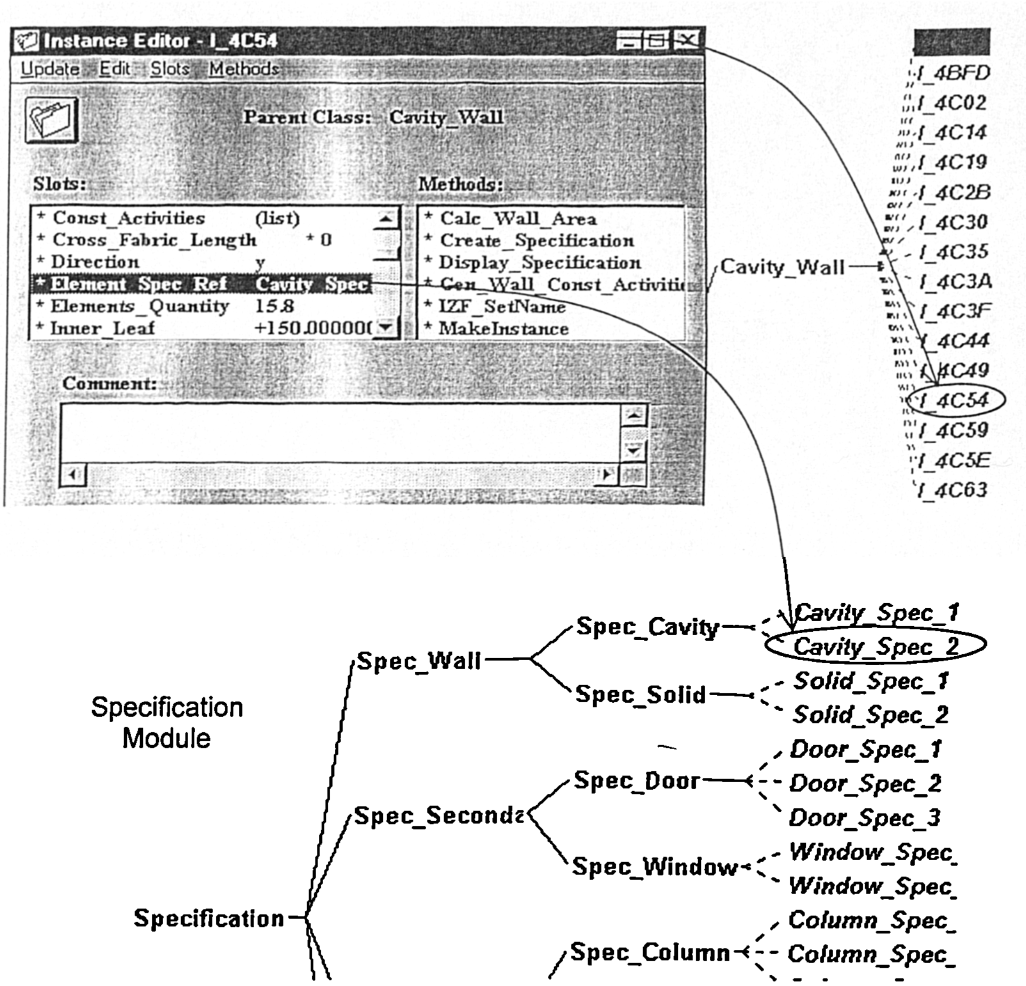
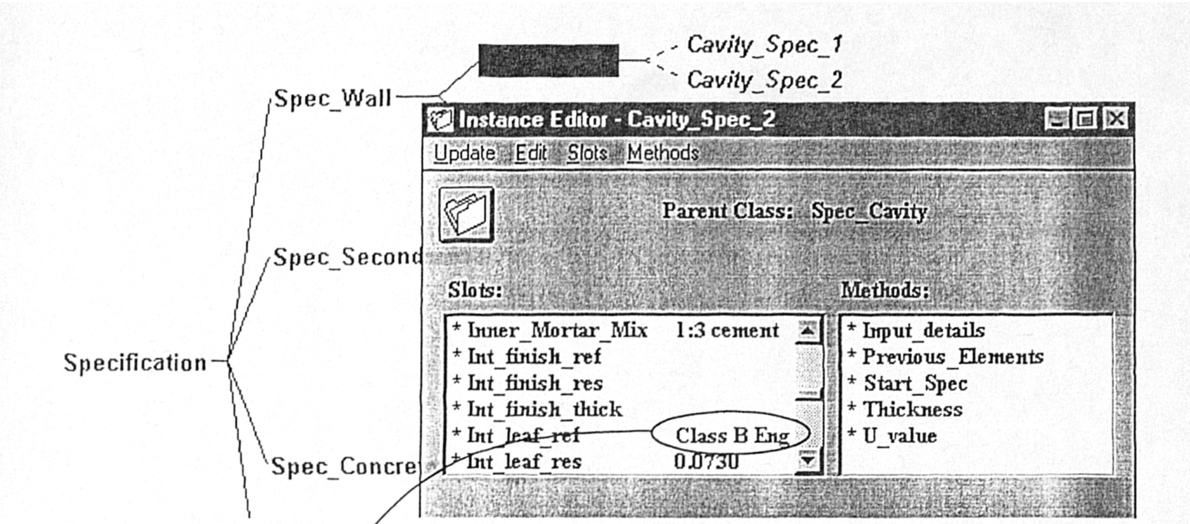


Figure 9.18: Cavity wall instance which is cross-referenced with the specification module

Using this method, specifications are not populated into the building element objects. This reduces data redundancies which could create problems when sharing the object information with other applications modules. The specification module stores all the related specifications in instance's slots while some of the static data is stored in external dbase format file. Figure 9.19 show details of a specification instance and its related static data, which is stored in a dbase format file.



MAT_CODE	MATERIAL	CONDUCT	RESITIVITY	THICKNESS	COST
Common 100	common half brick wall	0.000	0.0000	0.100	9.59
Common 215	common one brick wall	0.000	0.0830	0.215	19.60
Common 325	common one & half brick wall	0.000	0.0830	0.325	29.01
Common 440	common two brick wall	0.000	0.0830	0.440	38.38
Common 100	common half brick wall	0.000	0.0830	0.100	9.43
Common 215	common one brick wall	0.000	0.0830	0.215	19.12
Common 325	common one & half brick wall	0.000	0.0830	0.325	28.90
Common 440	common two brick wall	0.000	0.0830	0.440	38.37
Class A Eng 100	eng (A) half brick wall	0.000	0.0730	0.100	19.87
Class A Eng 215	eng (A) one brick wall	0.000	0.0730	0.215	39.69
Class A Eng 325	eng (A) one & half brick wall	0.000	0.0730	0.325	59.75
Class A Eng 440	eng (A) two brick wall	0.000	0.0730	0.440	79.01
Class B Eng 100	eng (B) half brick wall	0.000	0.0730	0.100	16.61

(Part of brick and block database file)

Figure 9.19: Detail of specification instance and specification database

iii) Define topological relationship

The identification of design element's association in the CAD system, which has been mentioned earlier, only provides the associated information. For example, a cavity wall is associated with a door, a beam is associated with a square column, etc. This information is not sufficient to provide the topological details which are needed, for example by a construction planner, i.e. it does not provide information such as, whether a beam is supported by a square column, a cavity wall is attached to a square column, etc. Therefore, a knowledge base should be developed to determine this information and the store them into the right object.

The knowledge base for the topological relationships between the building elements are set as follows:-

- A wall is attached to a wall (solid wall and cavity wall)
- A wall is attached to two columns
- A beam is supported by two columns
- A slab is supported by at least four beams
- A column is supported by another column
- A column is supported by a pad footing
- A window is embedded in a wall
- A door is embedded in a wall

A special function has been developed whereby instances (objects) which have the associated elements (attached from the CAD systems) will be scanned. The instances which fall into this list are sorted in a new slot, such as 'attached to', in the instance which has this topological relationship. The same procedure is then performed for other instances. Figure 9.20 illustrates an example of a cavity wall instance "I_4C54" which has been created with a topological relationship "attached to" slot and the slot contents are associated with column instances, i.e. "I_1718" and "I_1623". The instance "I_1623" associated slot contains three values, i.e. including the instance "I_4C54" which is referred to.

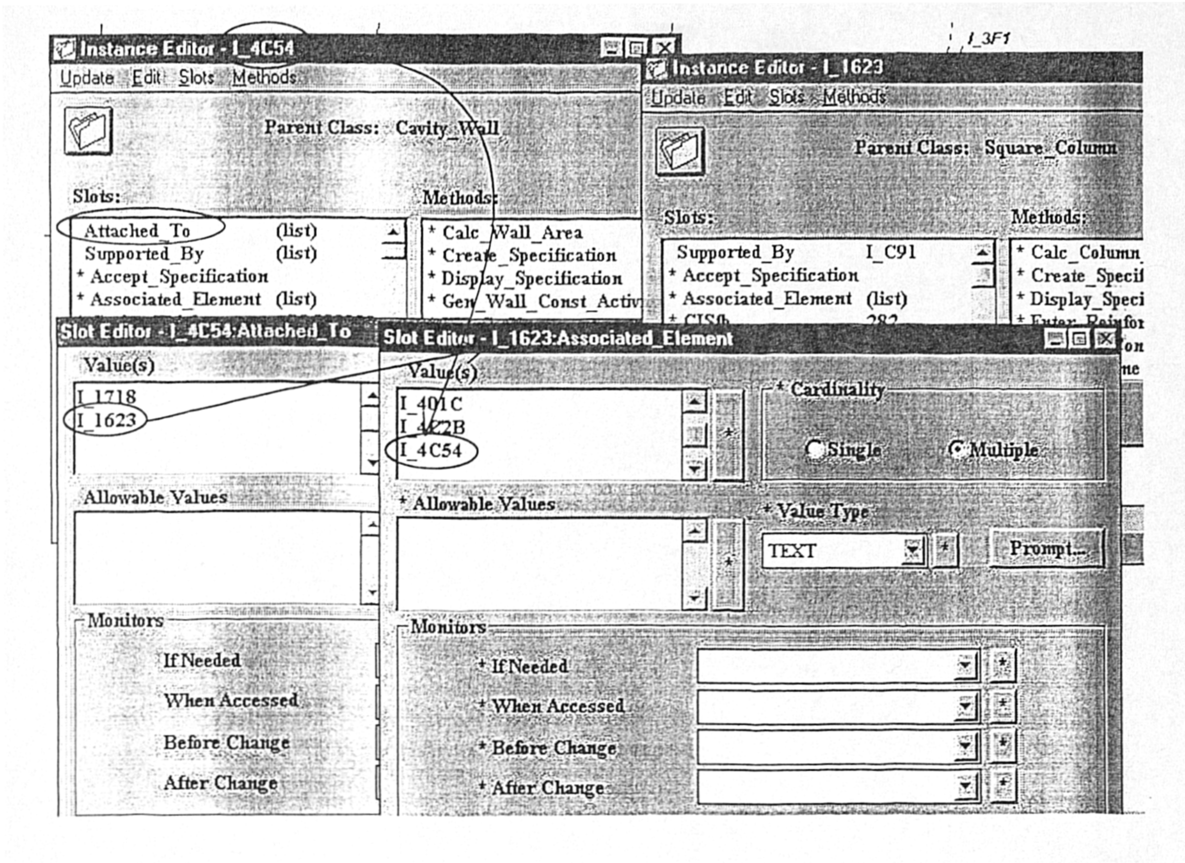


Figure 9.20: A cavity wall instance with "attached to" slot and their contents

Figure 9.21 shows another example of topological relationship in which a "supported by" slot has been created. In this case, a topological relationship between a square column "I_1623" which is "supported by" and another square column "I_C91".

iv) Check beam type

As mentioned earlier in Chapter 8, in order to support multiple views, i.e. the designer's and constructor's views, two separate models have been developed. The "building elements" data module contains the generic view of building elements such as columns, and beams which support the constructor's view. In order to support a designer's view a "building design" data module is developed which extends the design elements to include analysis and design information. In this study, only beams are considered in their two structural types; "simply supported and continuous".

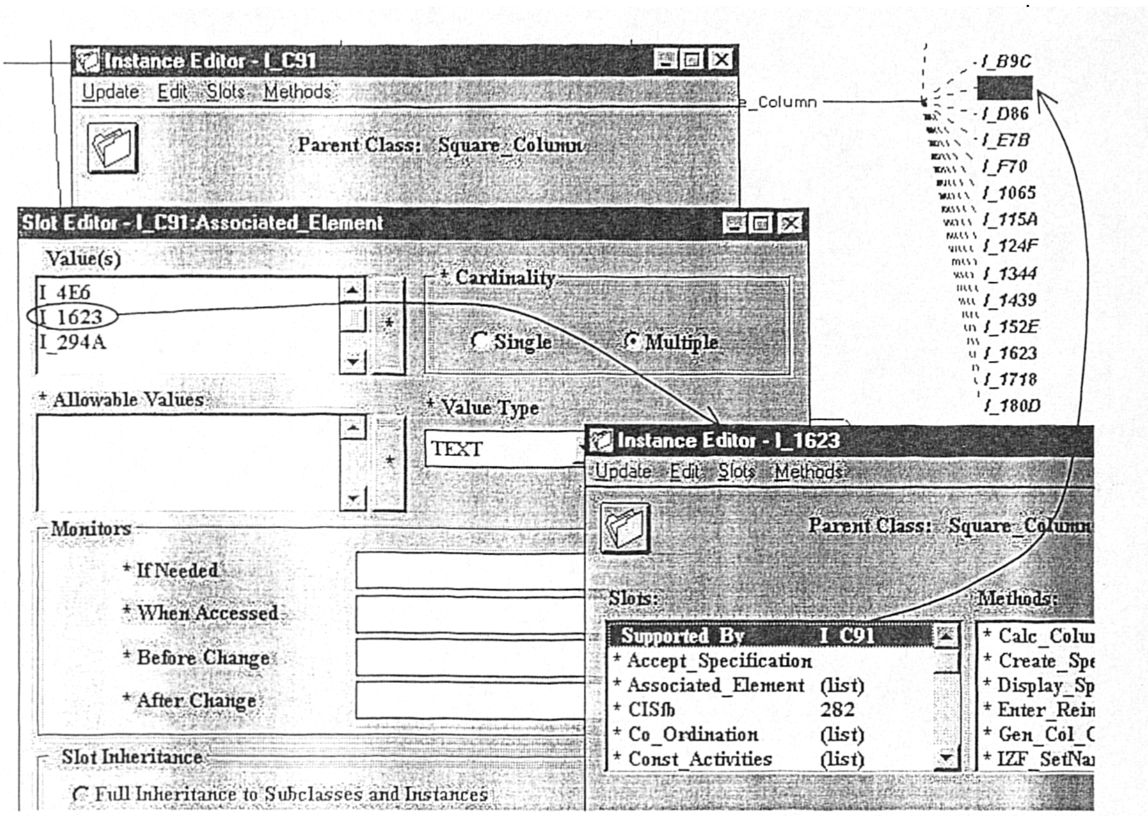


Figure 9.21: A square column instance which is "supported by" other square column

In this case, when a single beam instance is created (e.g. I_1985), it is cross-referenced to the "building design" data module as "SSB_I_1985" which means a simply supported beam. Figure 9.22 illustrates this case.

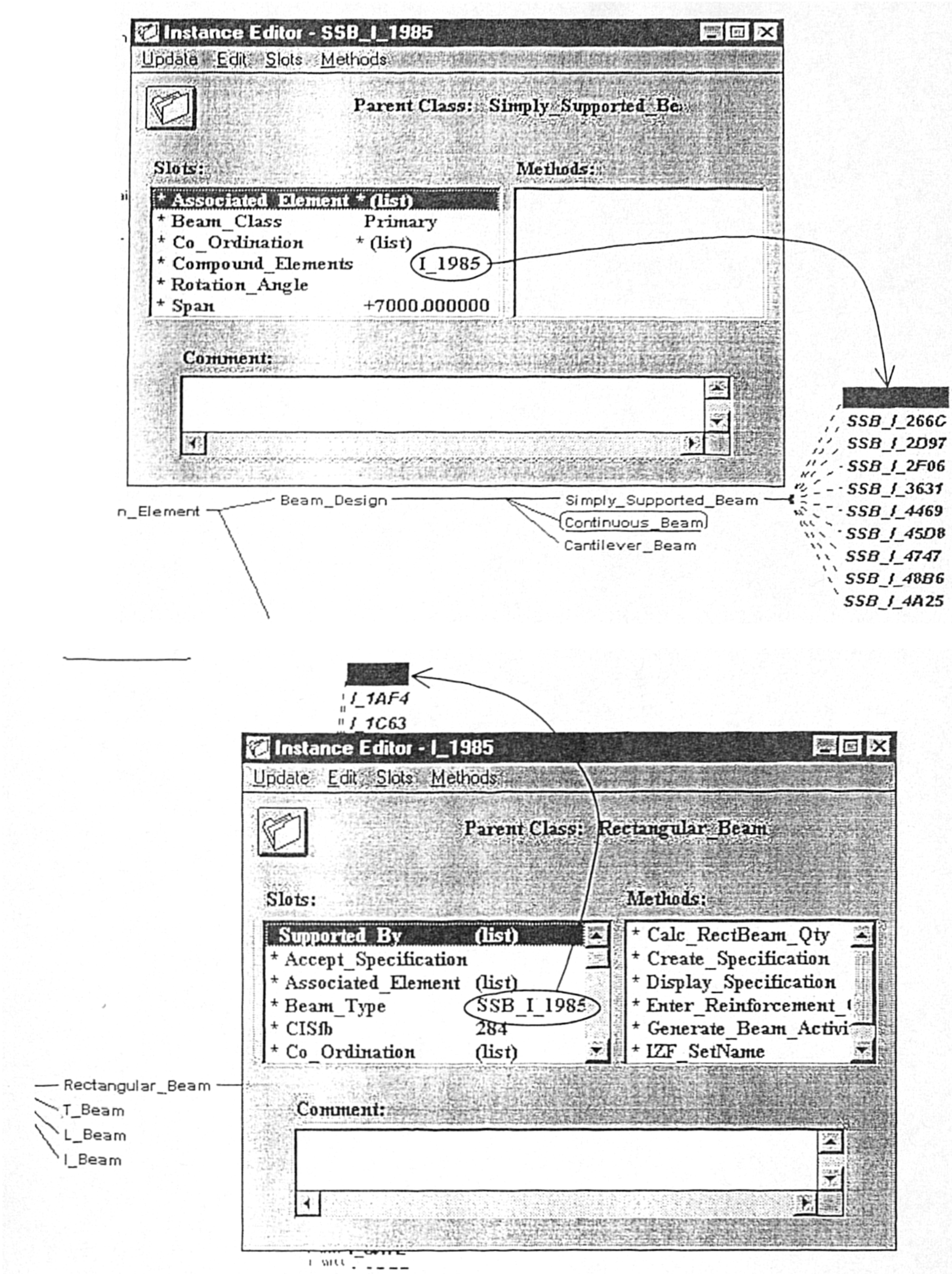


Figure 9.22: Relationship between a simply supported beam instance with actual instance in the "building elements" data module

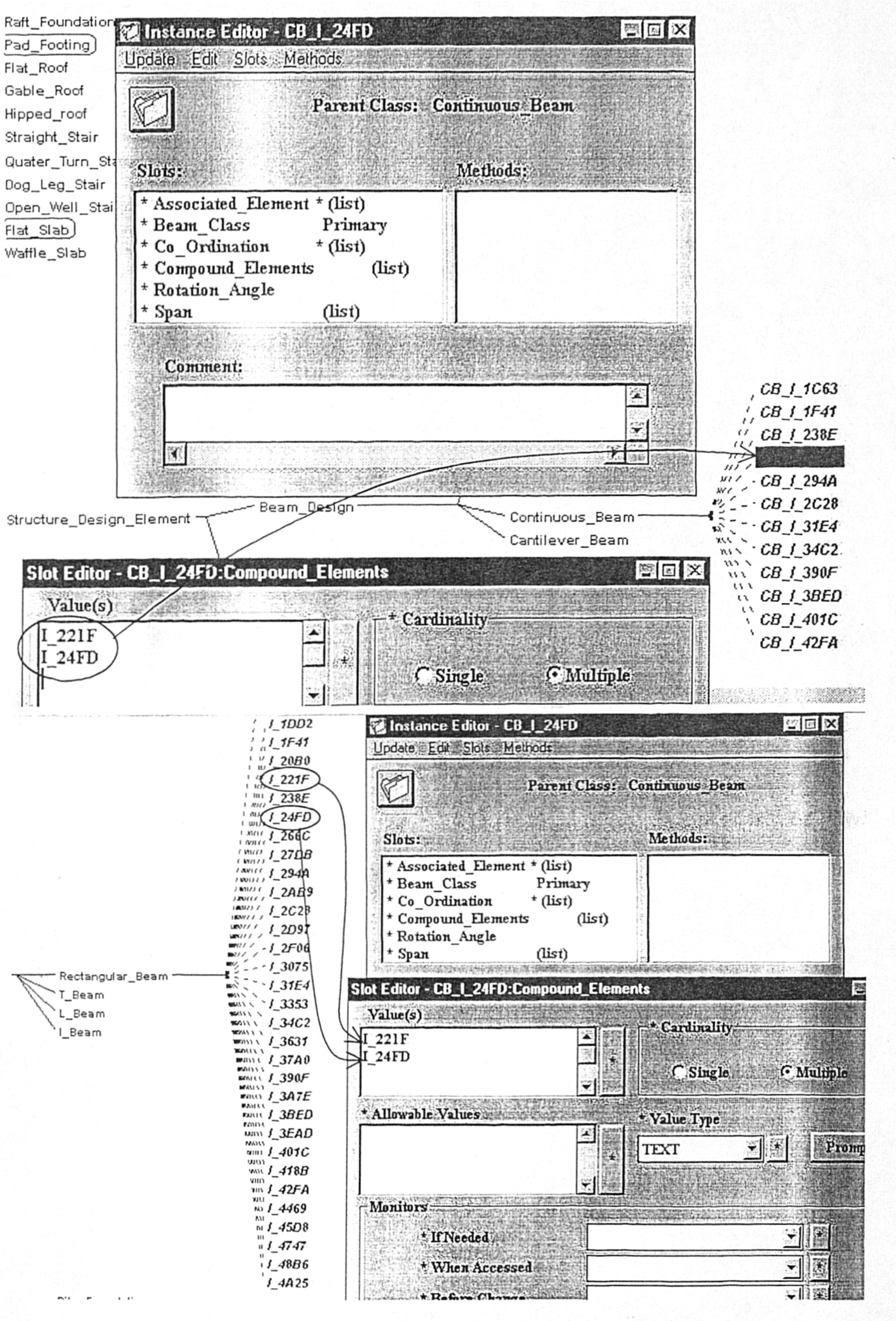


Figure 9.23: Relationship between a continuous beam instance with actual instance in the "building elements" data module

In a case where two beam instances are created (e.g. I_24FD and I_221F) where their directions are in line with each other and one of their co-ordinate is similar, a new instance "CB_I_24FD" is created to represent a continuous beam and is associated with beam instances, I_221F and I_24FD. Figure 9.23 shows an example of this case.

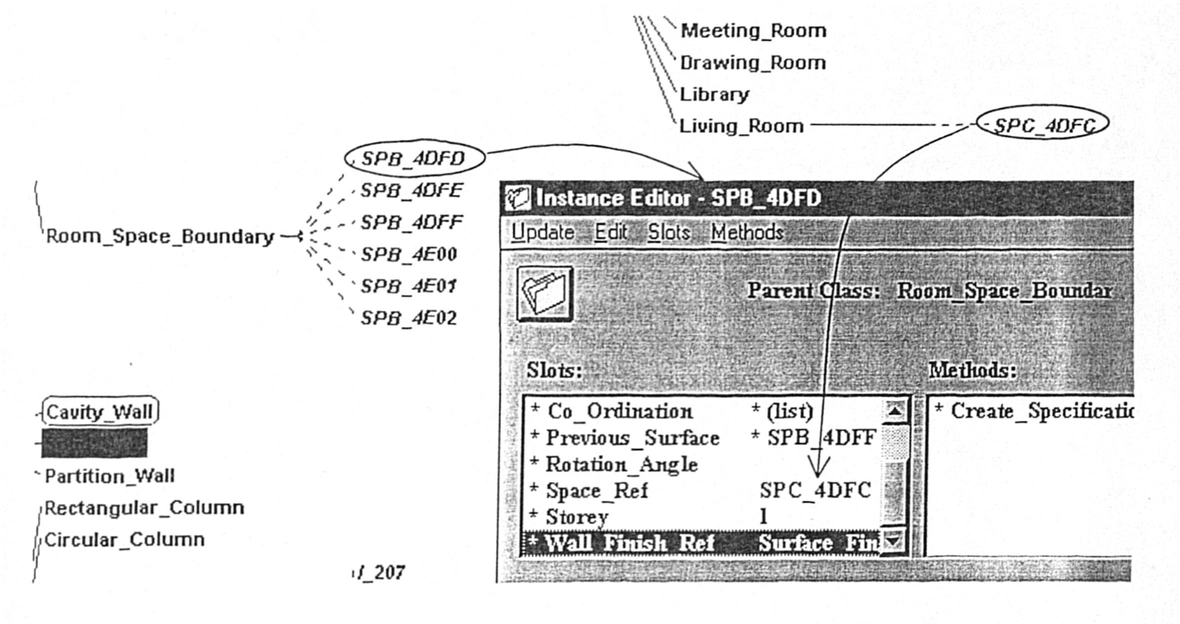


Figure 9.24: Space boundary with wall surface finishes specification

9.3.2.2 Space analysis

The space objects, which have been created in the CAD system, cannot be populated with vast information such as material specifications, etc. This is due to the limitations of CAD database and the complexity of storing further information. Therefore, objects are created in the object space hierarchy to overcome this problem. The space information and the space boundaries are stored in separate objects where the space object holds the information such as space area, floor finishes, etc., while, the space boundary object holds information such as space boundary, wall surface finishes, etc. Figure 9.24 shows a 'space boundary object', which has the wall surface finishes and Figure 9.25 shows a 'space object' which has the floor finishes. Details of both surface finishes specification are stored in the specification module.

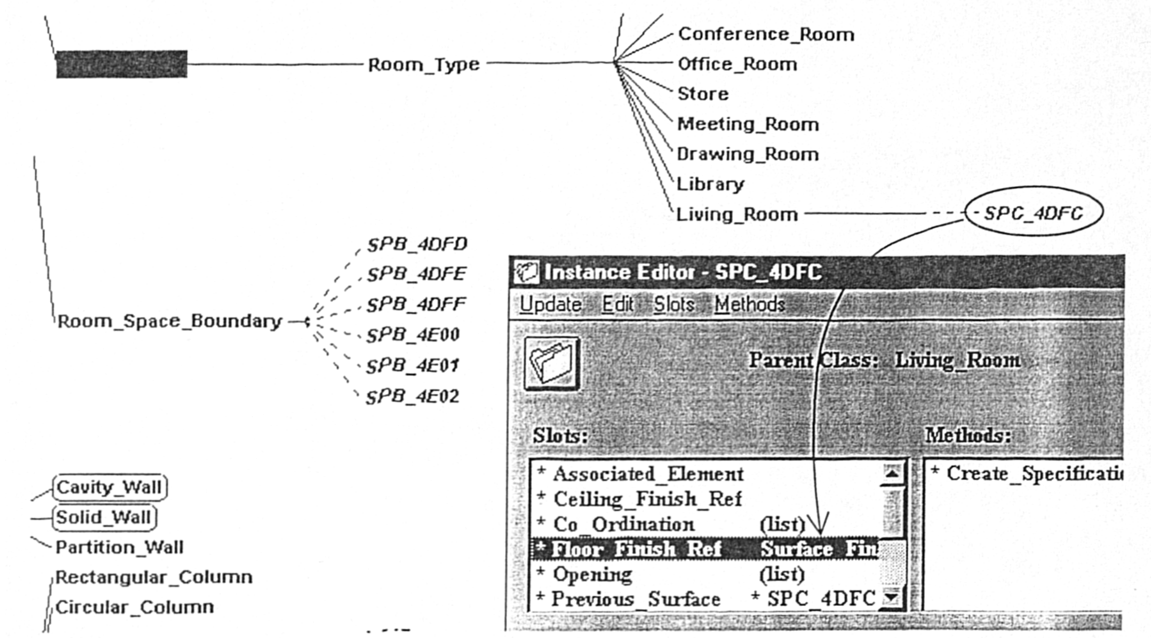


Figure 9.25: Space object with floor finishes specification

The functions required for analysing the space have been itemised as follows (Figure 9.3):-

i. Create space object

ii. Define space associated elements

iii. Define space boundaries

i) Create space object

After the process of defining and analysing the space in the CAD systems, the defined spaces are transferred and created as space objects in the object-oriented environment. The process of creating the space objects is similar to that of the building element object. The space object name is the handle name which is retrieved from the space entity in the CAD systems. Since KAPPA cannot handle instance name starting with a number an extension of "SPC_" is used.

Once an object is classified as a space object, a space object is created as an instance in the space data module under the defined class. For example, when a user defines a space name as a "living room", an instance is created at the "living room" class as shown in Figure 9.24. The process of defining the space name is controlled by the building space data module whereby only the classes, which are defined in the "Room_Type" object are selected. This is to ensure no error occurs when creating a space object in the project model.

ii) Define space associated elements

During the analysis of the space in the CAD system, the building elements, which are associated with the defined space, are captured and associated with the space entity. These associated elements are then passed to the project model after the space instance is created whereby each associated element is been separated either as a space separator or a space opening. Space separator includes wall and flat slab, while space opening includes windows and doors. Figure 9.26 shows an example of a list of space separators and space openings for a space object.

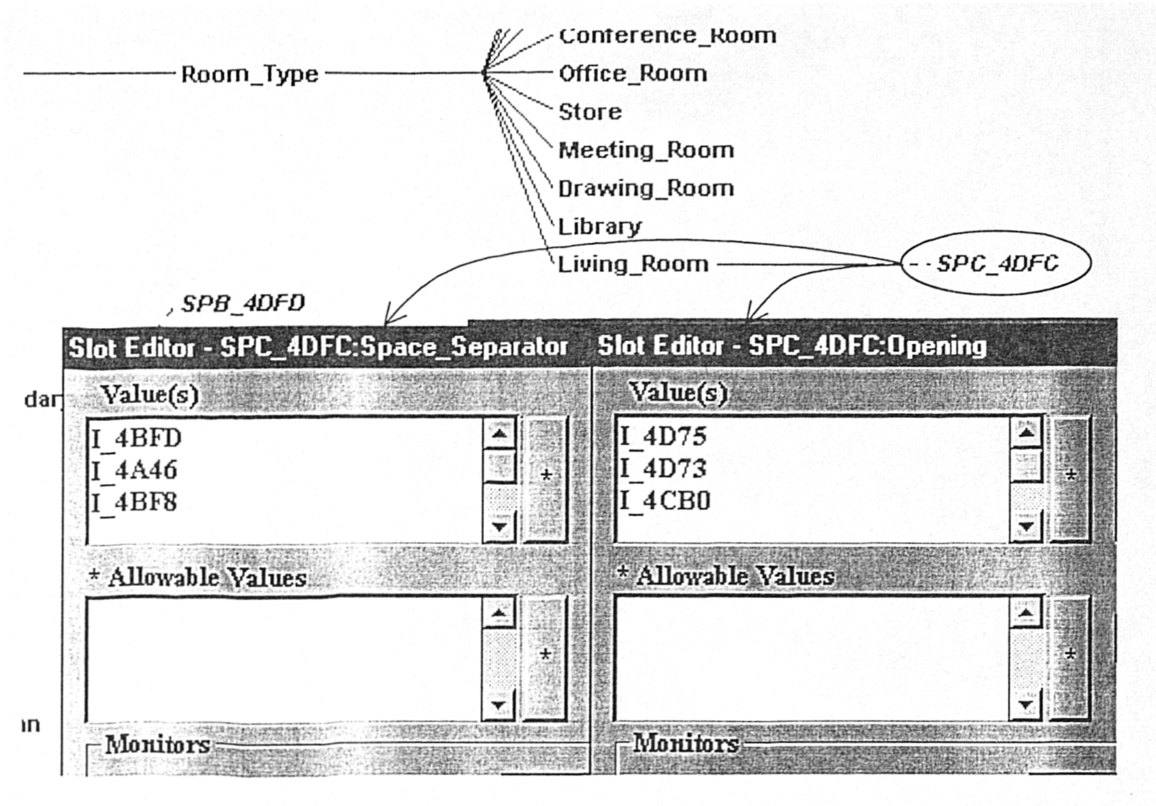


Figure 9.26: Space object with space separator and space opening

Since the associated elements captured for the space separator include walls, windows and doors, excluding slabs, they have to be sorted in order to include slabs in the space separator list, while the window and door in the 'space opening' list. This function is developed using an algorithm below and Figure 9.27 shows an example of a sorted space separator which includes cavity wall and flat slab.

Get the associated elements from CAD

Check each element

    If the element is a "Door" OR a "Window" Then set as "Space Opening"
    If the element is a "Cavity Wall" OR a "Solid Wall" Then set as "Space Separator"
    Get the associated elements
    If the associated element is a "Flat Slab" Then include this element in the "Space Separator" list

End of checking

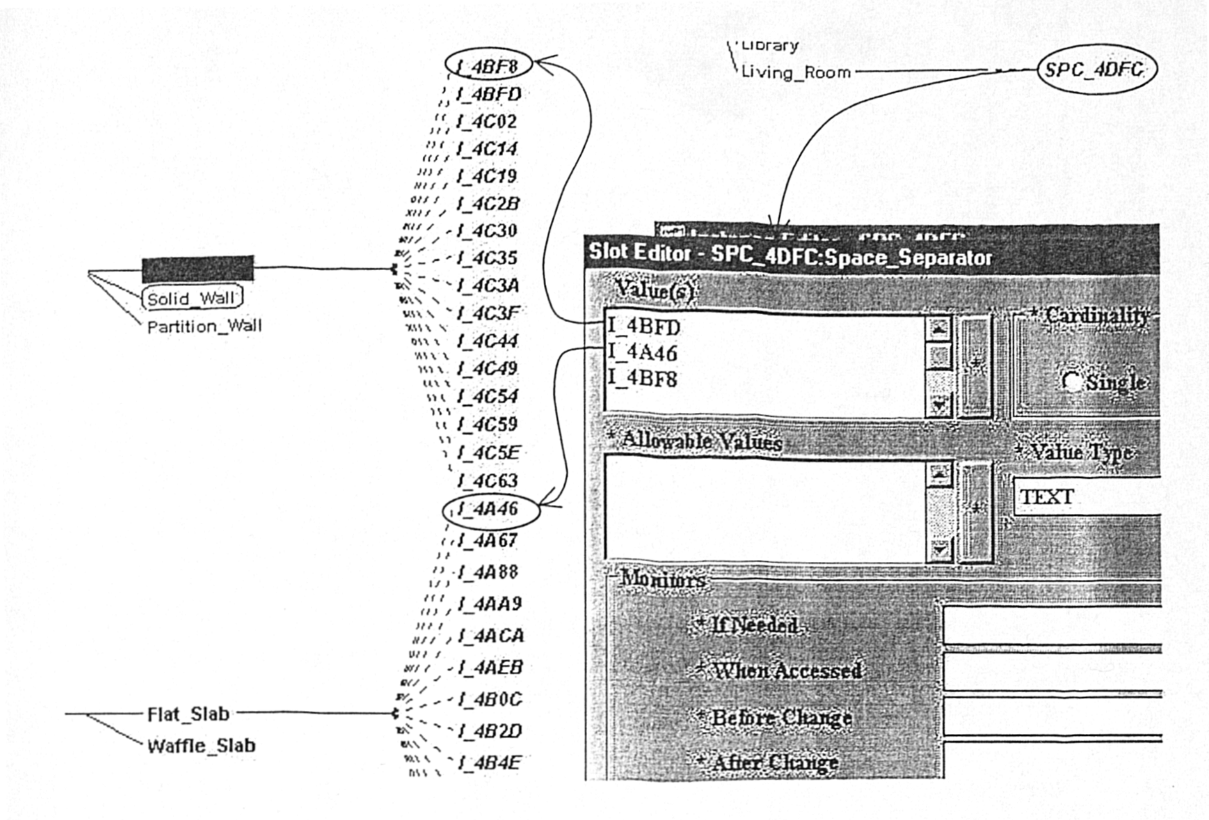


Figure 9.27: Space separator including cavity wall and flat slab

iii) Define space boundaries

The space boundary's objects of a defined space also need to be created in the project model. As mentioned earlier, such objects are required for defining other related information with the space boundaries such as surface finishes, surface area, space separator, etc. In order to create the space boundary's objects, each space boundary which is captured while defining a space in the CAD systems will be created at the 'Room_Space_Boundary' object/class. Like the 'space object', the space boundary's objects will be created with an extension of "SPB_" plus the space boundary handle name. All the space boundary's objects will be cross-referenced with the 'space object' as shown in Figure 9.28.

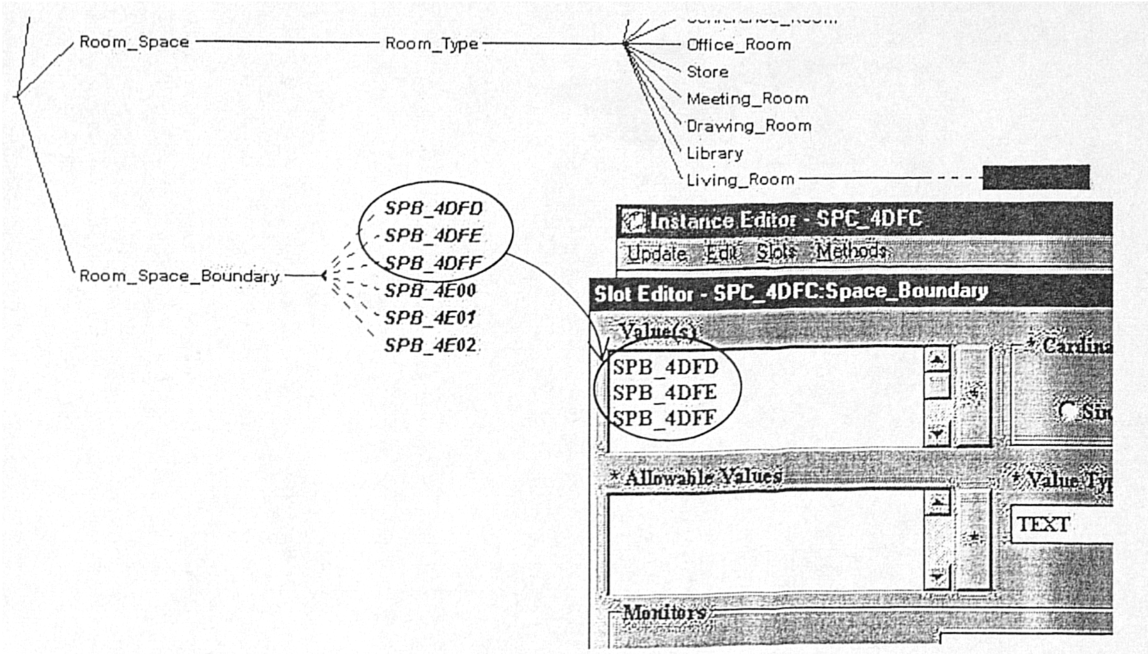


Figure 9.28: Space object which is associated with space boundaries objects

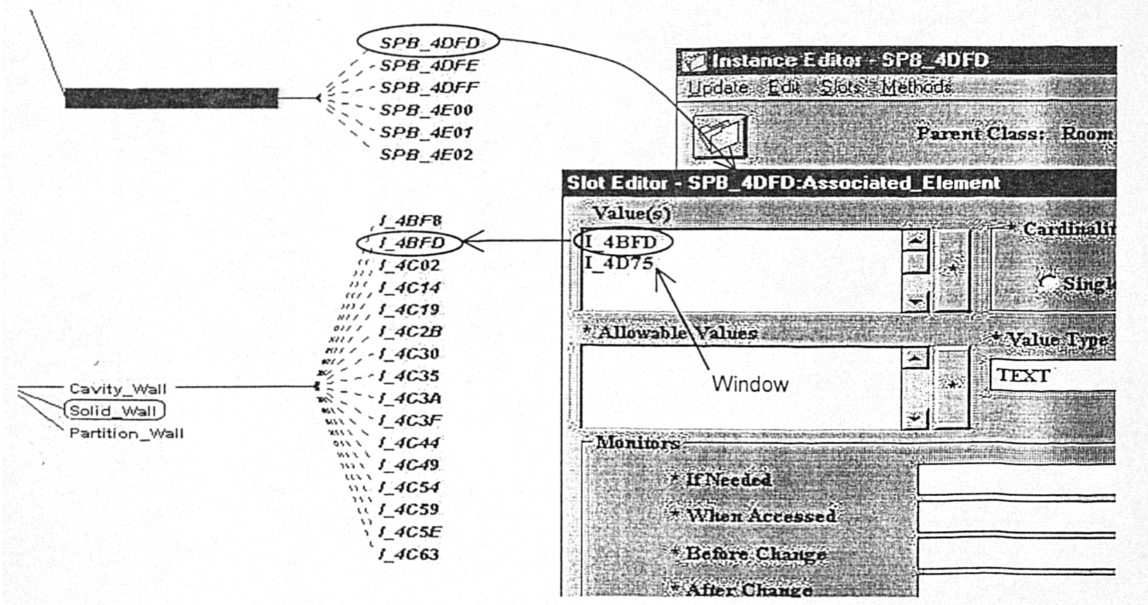


Figure 9.29: Space boundary which is associated with space separator (cavity wall)

Space boundary is also associated with space separator and space opening where the total surface area can be determined. Figure 9.29 shows an example of a space boundary 'SPB_4DFD' which is associated with space separator (cavity wall) 'I_4BFD' and space opening (window) 'I_4D75'. If the dimensions of both a cavity wall and a window are used then the surface finishes area can be determined.

9.4 System's application

CAPE was developed to support other applications such as construction planning, site layout planning and estimating in the SPACE integrated environment. CAPE is not a construction application as such and therefore it can only be applied when the project model is populated with design and other related information. This is done as soon as the user triggers the customised AutoCAD-AEC™, i.e. by defining building elements such as walls, windows, doors, etc. Once the building element data module is populated with this information, other applications will be able to access it. CAPE operations can be illustrated in an application of an object through its life cycle whereby an object is created, amended, supplemented with other related data and used by other application modules in the modularised environment as described in Chapter 6 and 7.

9.4.1 Application of an object through their life cycle

The concept of object's definition and object's life cycle described in Chapter 7 have been utilised starting from the creation of an object to the usage of the object. Figure 9.30 illustrates this concept in three steps, i.e. a) create and amend object; b) supplement object with data; and c) use object. This concept illustrates how such object can be created, shared, manipulated and used by other applications.

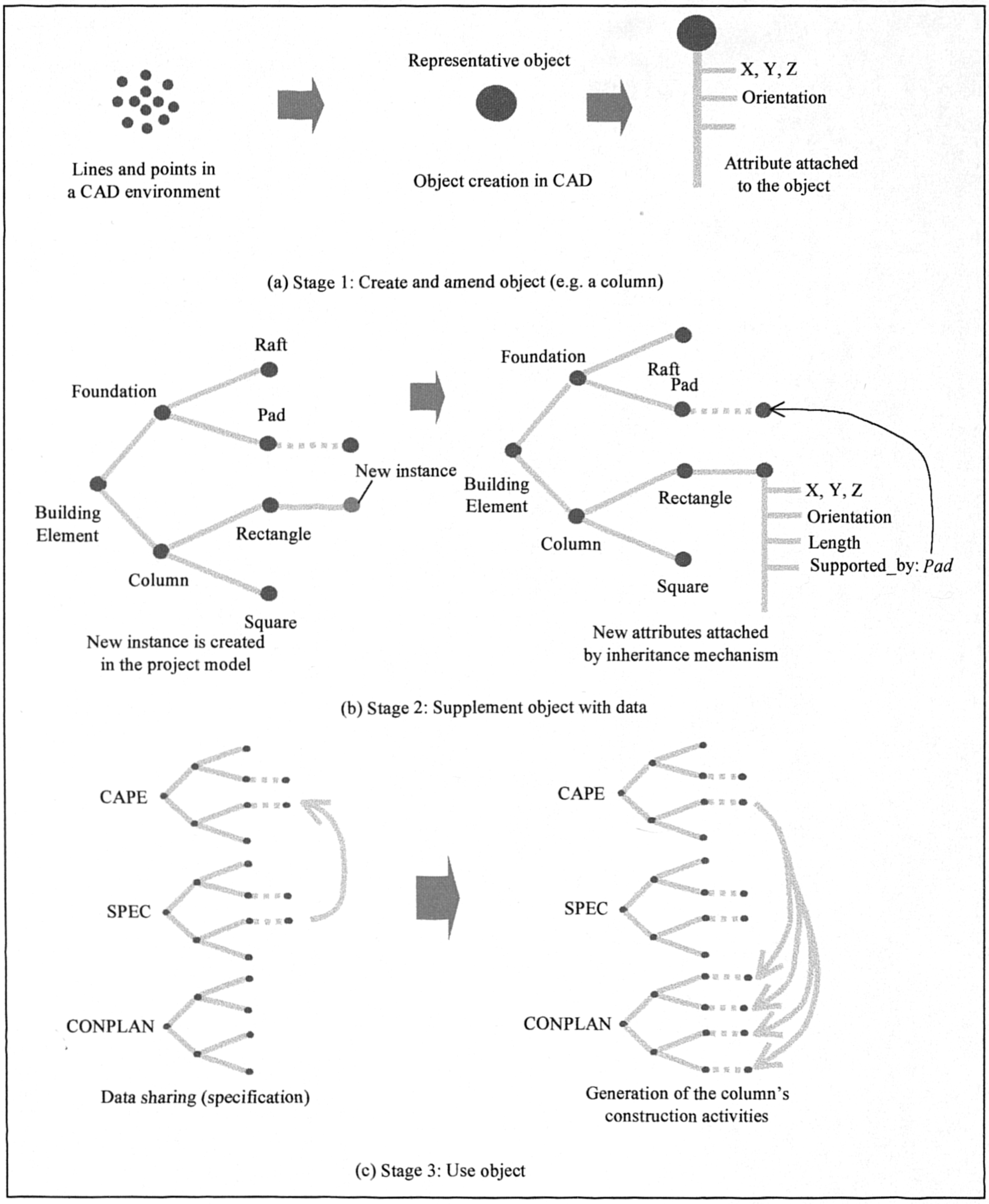


Figure 9.30: Application of an object through their life cycle

a) Create and amend object

Starting at the CAD package, if a design element, say a column, is drawn in a "wire-frame" system, it is represented as a number of drawing primitive entities, i.e. a group of connected lines. The Object Interpreter Engine (OIE) automatically captures these primitive entities and creates a representative object to represent the element in the CAD database. The representative object is then treated as an object where a number of attributes is attached to it such as x, y and z co-ordinates, orientation, length, etc. The representative object has its own standard graphical file in order to represent itself when required by other graphical applications such as CAD and VR. At the end of this stage, the object will know its type, shape, location, and other related geometric information. Figure 9.30(a) depicted this stage of the object's life cycle.

b) Supplement object with data

In this stage, the representative object is dynamically transferred to the project model. It attaches itself to the right class in the elemental hierarchy, i.e. creating a new instance for the column at the building elements module where the column is treated as part of the design developed so far. As shown in Figure 9.30(b), a pad foundation instance has also been created. The new instance, i.e. the column, inherits all its properties and behaviour from its parent class, utilising the object-oriented features of the environment. At this stage, the object knows its identity, structural function and other properties that it need to know about such as supported-by and adjacent-to.

A design application has been developed to automatically identify the topological relationship between the design elements such as supported-by element and the associated with element. In this stage, the column object automatically checks its associated elements and attaches itself to other objects, which are associated with it. In this case, the column object is supported by a pad foundation and its cross-reference with the pad foundation object as shown in Figure 9.30(b). This application which is built in the building elements data module is automatically triggered off when the new instance is created. This building elements module is only one part of the modularised environment. The whole environment contains other application modules such as SPEC, CONPLAN, etc. as shown in Figure 9.30(b). Other application modules have been described earlier in Chapter 6.

c) Use object

When the newly created instance declares itself in the environment, the SPEC module automatically responses by defining the best specification for the object. If a similar specification for a column is already exist in the environment, the SPEC module cross-reference the new column object with the existing specification object. Otherwise, it creates a new specification object and cross-reference it to the new column.

When the planning application is triggered off, it generates the required construction objects as required by the plan and then cross-reference them to the design object. All the related information in the design object, which is required by the construction planning application, becomes available. The construction objects are then transferred to the construction planning application package for generating the construction plan. Figure 9.30(c) depicted this stage of the object's life cycle.

This example only shows the use of object with the SPECIFICATION and CONPLAN application modules. Other application modules such as EVALUATOR, INTESITE and CONVERT use the objects in the same manner as been described above.

9.5 Summary

This chapter has outlined an overview of the CAPE system development process as part of supporting the development of SPACE integrated environment. It highlights the CAPE system's architecture, implementation and application. The system's architecture briefly describes the system input, the knowledge base required by the system, and its output. The system's implementation describes in detail how it was developed and it is divided into two main parts, i.e. the CAD system and the object-oriented environment. The system's application on the other hand, briefly highlights the application of an object through its life cycle where the object is created, amended, supplemented with other related data and used by other application modules in the modularised environment, i.e. SPACE. The following chapter describes the demonstrations and experiments performed by CAPE within the integrated system.

Chapter 10

Demonstrating and Experimenting with the Prototype

10.1 Introduction

This chapter demonstrates the prototype's capabilities for achieving its objectives in the central core of integrated environments. The demonstrations illustrate how the information stored in the CAPE module are used by the other applications and viewed in virtual space. Finally, it highlights the procedure for experimenting with the prototype and validation of the results using the users/evaluators approach. By referring back to the hypothesis of this study, this chapter is only concerned with testing of the project information, their validity, availability and their support to other major downstream applications.

10.2 Demonstrating the prototype

As mentioned earlier, CAPE creates and provides essential data for other construction applications. However, the data which is held in the building element data module is of great importance to professionals and can be utilised in different ways. This research has developed a sample function to demonstrate few of the envisaged usage of data. These functions are mainly aimed towards the "integration" of the design. Figure 10.1 shows the main screen of SPACE with CAPE demonstration menu.

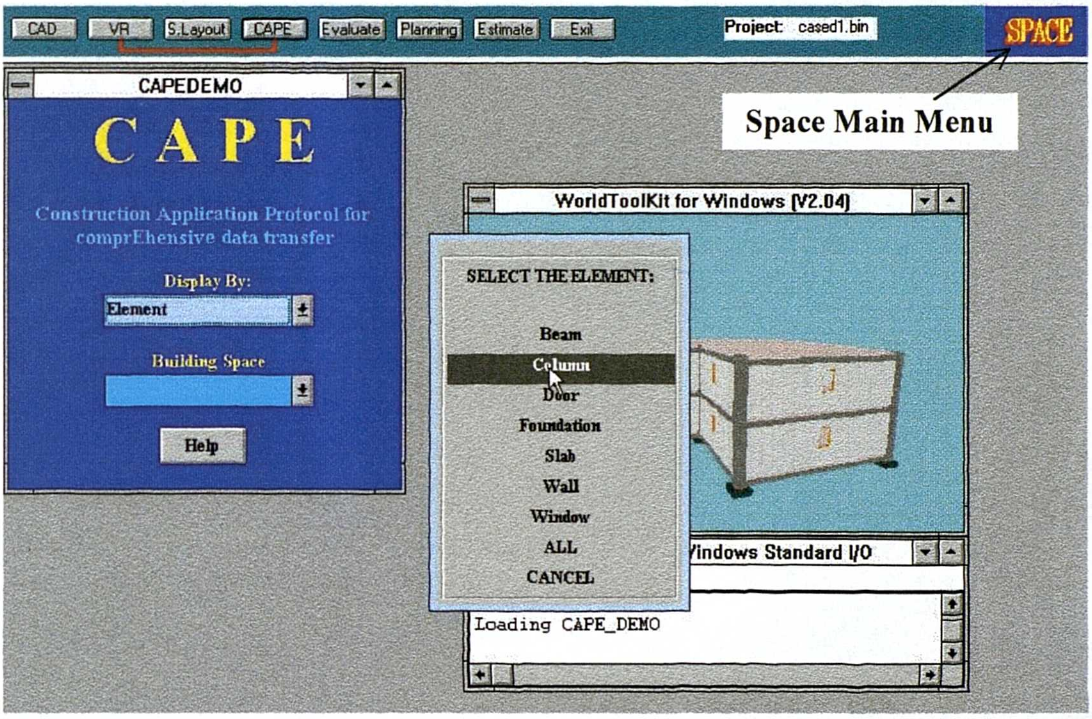


Figure 10.1: Main screens of SPACE with CAPE demonstration menu

The CAPE module is invoked by selecting the CAPE button from the SPACE main menu as shown in Figure 10.1. The VR application is simultaneously triggered off so that the design information can be manipulated in virtual space. The prototype demonstrations are divided into three main parts as follows:-

1. Selection of elements;
2. Topological relationships; and
3. Space definition.

10.2.1 Selection of elements

The design elements, which are populated into the project model, are attached to the right class of the elemental hierarchy. For example, a cavity wall created in the CAD system is created/attached as an instance, to its parent class/object, i.e. cavity wall class/object. Once all design elements are transferred into the building element data module, the design can be interrogated. In this demonstration, the design elements are shown according to material's specification, storey and total cost.

a) By material specifications

Using CAPE query window, the user can access the required information in a graphical and textual format. For example, a column of a certain specification, i.e. sulphate cement is selected for viewing in VR as shown in Figure 10.2. The related columns, which use the sulphate cement, will be displayed in VR. Its associated textual information is printed in a report. This demonstration illustrates how the material specification of a certain design elements can be manipulated when required.

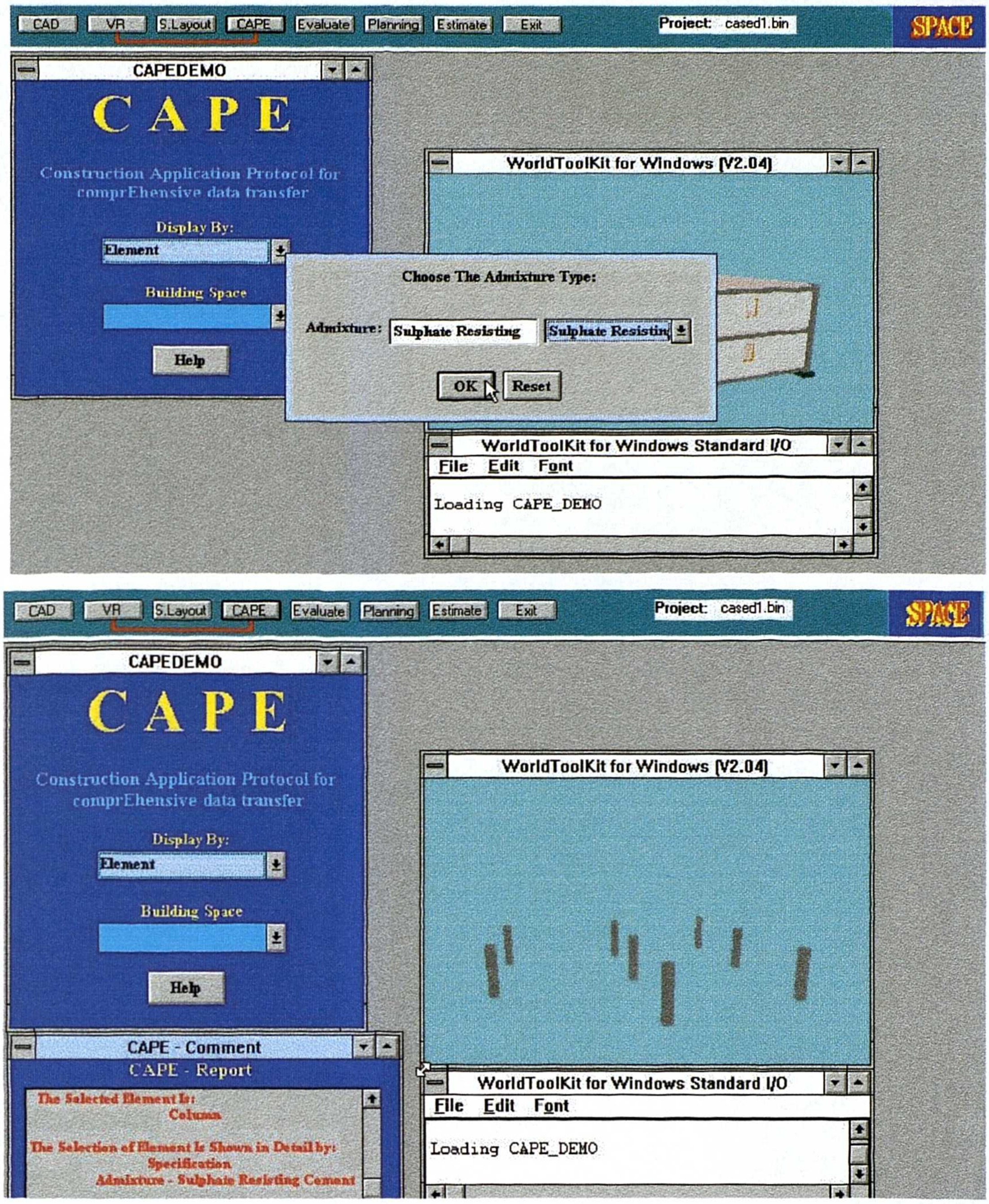


Figure 10.2: Elements display by material specification

b) By storey

CAPE gives users greatest flexibility to rapidly examine different aspects of the building. Instead of selecting and viewing the design elements using material specifications as mentioned earlier in (a), design elements can also be viewed according to the storey/level of the building. For example, all beams in the building can be viewed, i.e. by selecting the beam element for display using 'All storey' as shown in Figure 10.3. This demonstration gives the structural engineer a high level view of part of the building's structure where they can examine the structural function, analyse and design any other structural elements.

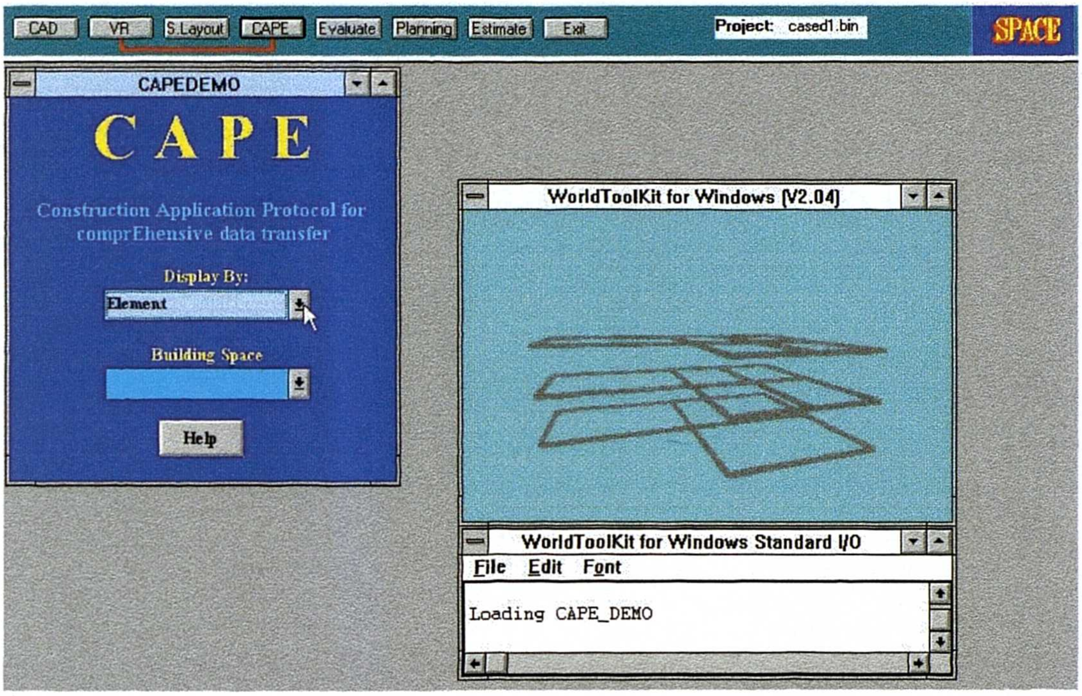
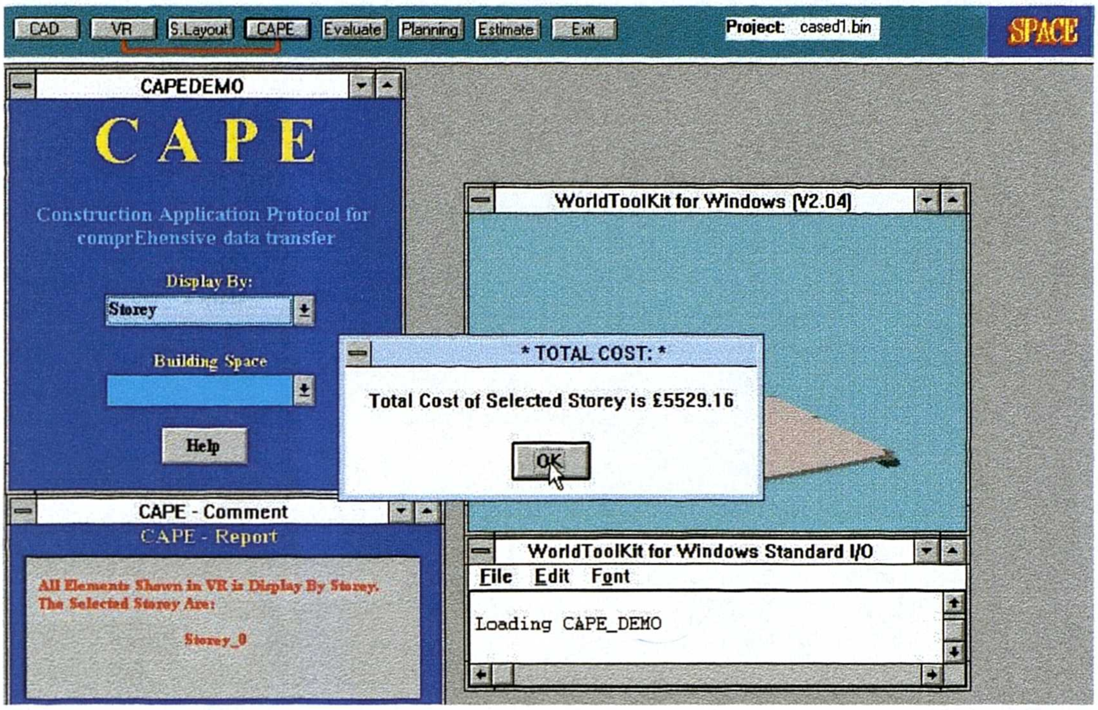


Figure 10.3: Selection of beams in all storey/level

c) Total project cost

The cost for a single group of elements can also be determined using the query window. For example, the cost of the foundation, i.e. at storey 0, can be viewed in this demonstration as shown in Figure 10.4. The cost displayed is the total project cost for a single group of elements, i.e. the total elemental and construction cost calculated in other application modules. This demonstration also shows how other application modules such as EVALUATOR and CONPLAN can be interrogated by CAPE to access the elemental and construction cost.



<u>Figure 10.4</u>: Total cost of selected storey

These queries are just a few examples of how CAPE can be interrogated. Users are also able to walkthrough the virtual design and interact with any elements for further information when required. The ability to visualise and analyse the design elements enables designers to optimise the design thus, reducing the likelihood of any later problems.

10.2.2 Topological relationships

The CAPE model can also be interrogated in the virtual space simply by clicking on any of the displayed elements. CAPE responses by displaying a number of suggested functions. For example, the structural function only allows the structural elements to be displayed as shown in Figure 10.5. The topological function highlights the relationships between the building elements such as supported by or associated with. Such relation can be interpreted graphically in VR to enable users to quickly and easily highlight unforeseen design problems. This is demonstrated by trying to delete one of the pad foundations. CAPE responses by shading/colouring all the effected elements (in blue) and printing their ID on the screen as a report. This is illustrated in Figure 10.6.

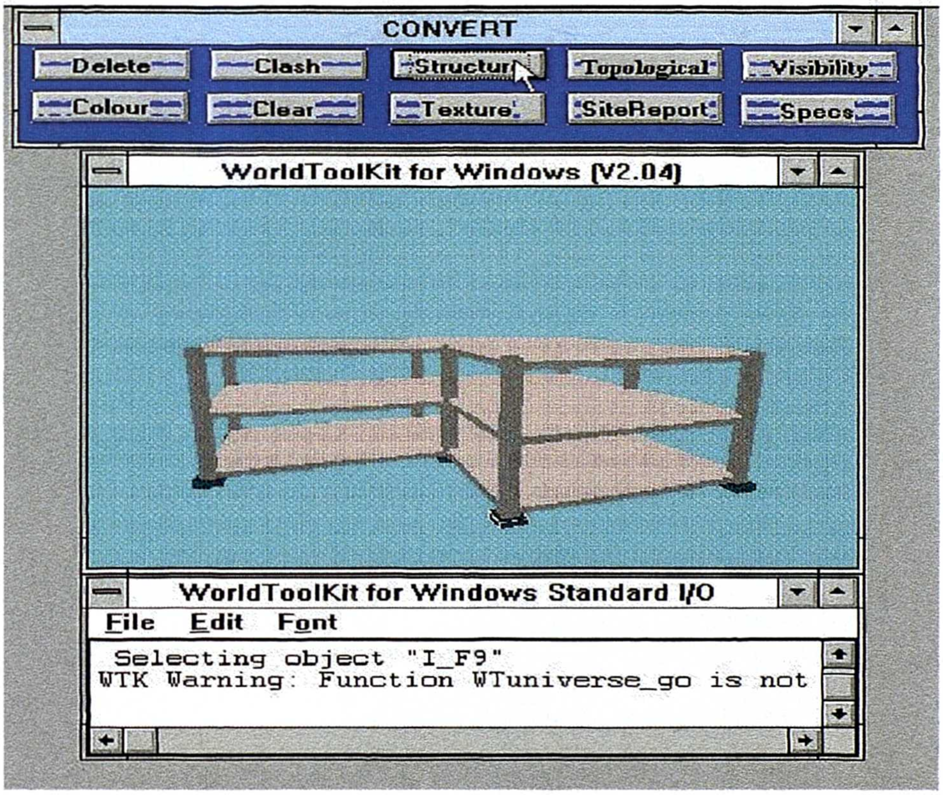
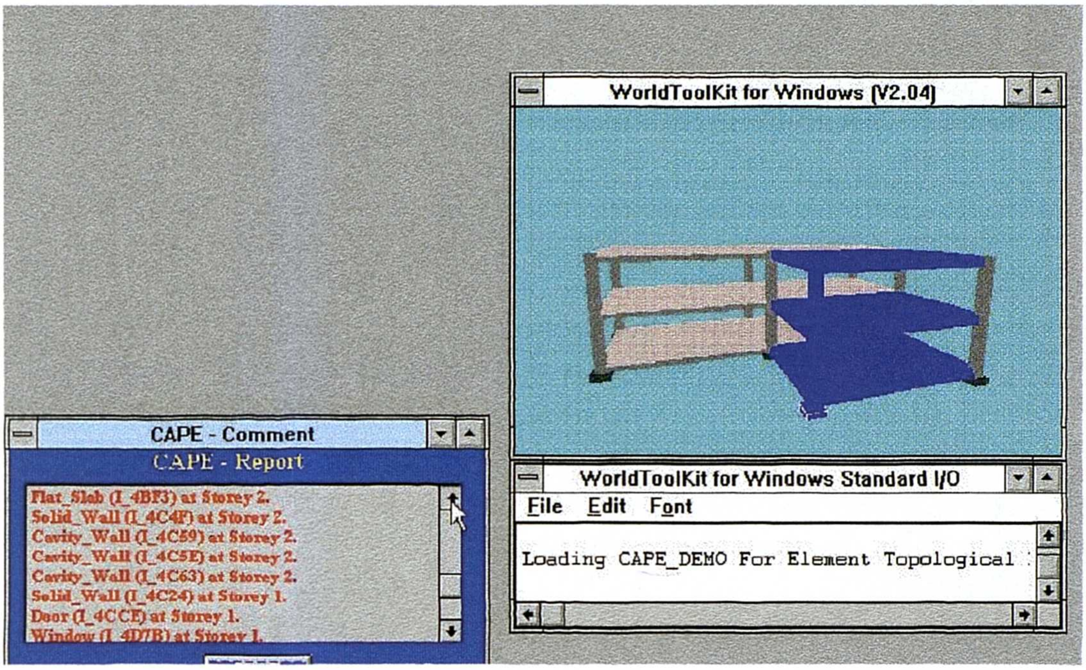


Figure 10.5: Structural elements of a building

The element, which is associated with other elements, can also be demonstrated. For example, when a column is selected, CAPE responses by shading/colouring (in blue) the associated elements with this column as shown in Figure 10.7. This information provides a high level view for the designer to check the topological relationship of any elements before delivering it to the construction planner.



<u>Figure 10.6</u>: Effected structural elements when a pad foundation is deleted

10.2.3 Space definition

A special application has been developed to define spaces from the building's plan. This application is developed in the CAD system where additional functions have been added to the AutoCAD-AEC^{TM} pull-down menu as shown in Figure 10.8. These two functions (Define Space and Specify Space Finishes) are used to define the space and space finishes where the space boundaries will be captured and later the right surface finishes will be attached to the space and the boundaries as discussed in Chapter 9.

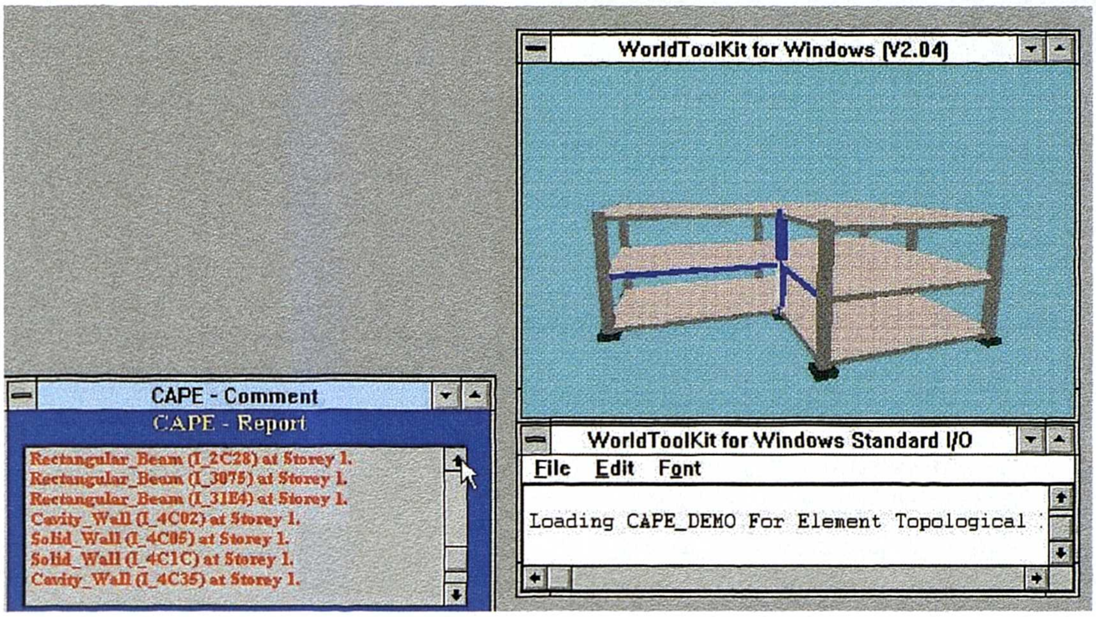


Figure 10.7: The associated structural elements with a selected column

In order to interrogate the defined spaces in the CAPE module where the space boundaries and the surface finishes are stored, the same query window of CAPE is used. Three different options have been developed, i.e. the space boundaries and the surface finishes for walls and floors as shown in Figure 10.9. In this demonstration, the CAPE query window is invoked from the customised AutoCAD-AEC™ side menu.

From the CAPE main query window, information associated with the defined space can be retrieved by clicking on either the space object or its relevant components. This is shown in two different demonstrations, i.e. by displaying the space boundaries and by surface finishes for the defined space. Figure 10.10 shows the space boundaries are highlighted for a selected space, living room, in which the textual information is also printed on the screen. Thus, this information shows that the defined space boundaries, which are stored in the CAPE module, can be visualised when required.

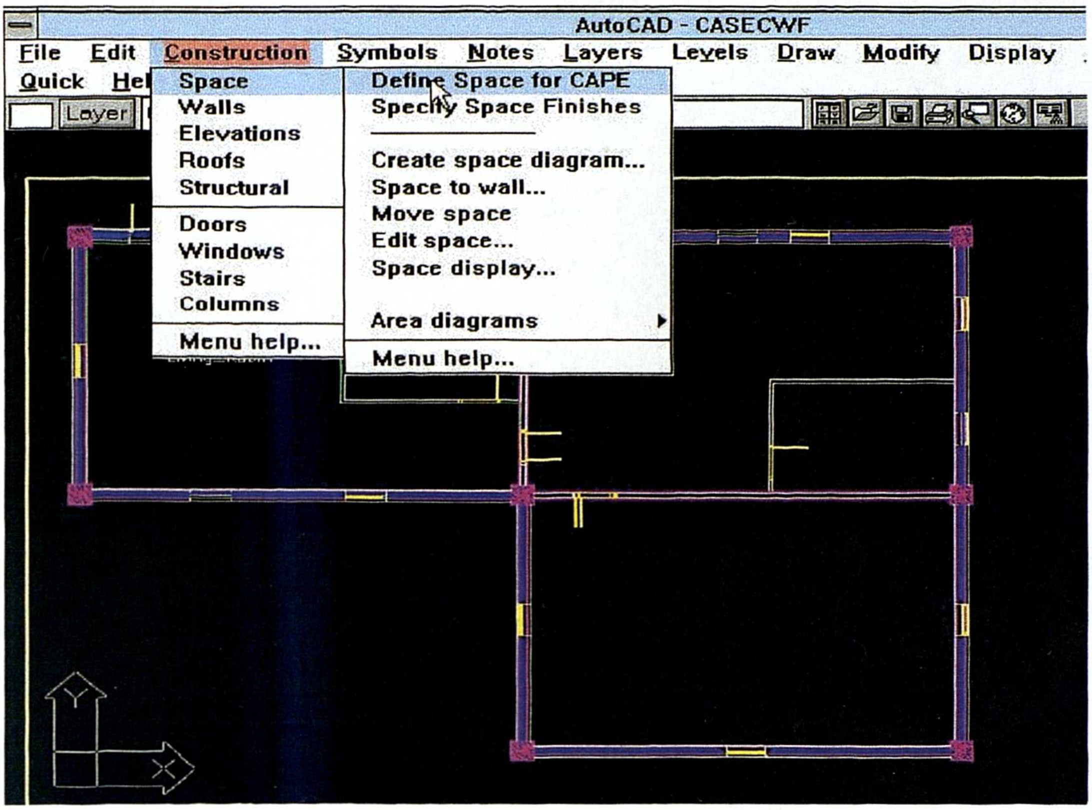


Figure 10.8: Main menu screen for defining a space in CAPE

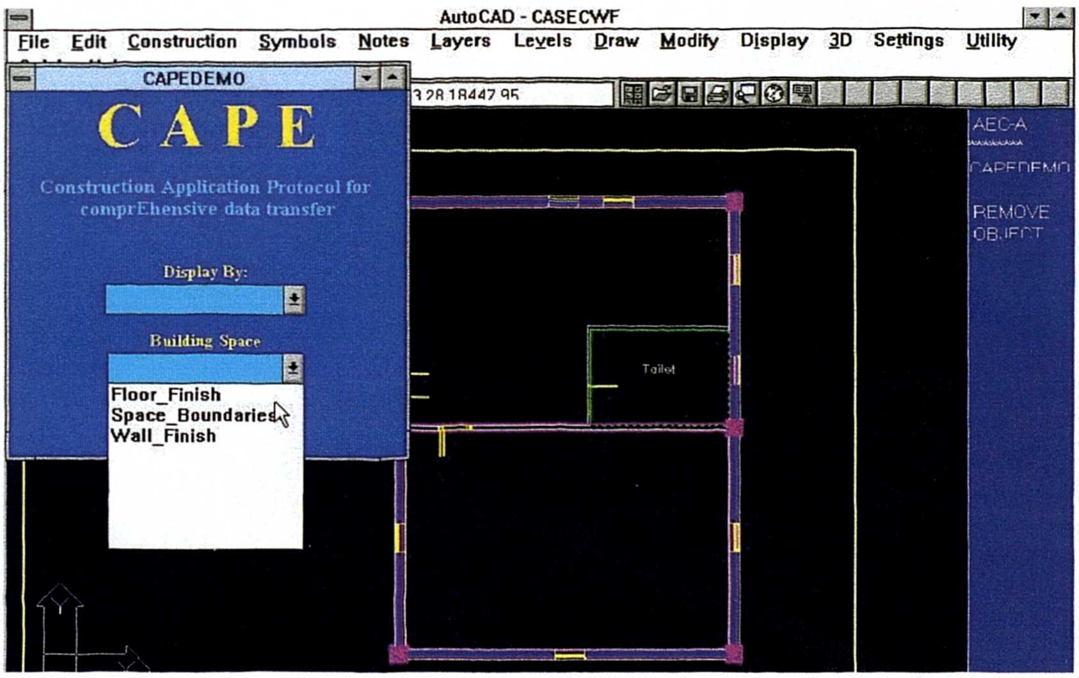


Figure 10.9: CAPE queries window for space demonstration

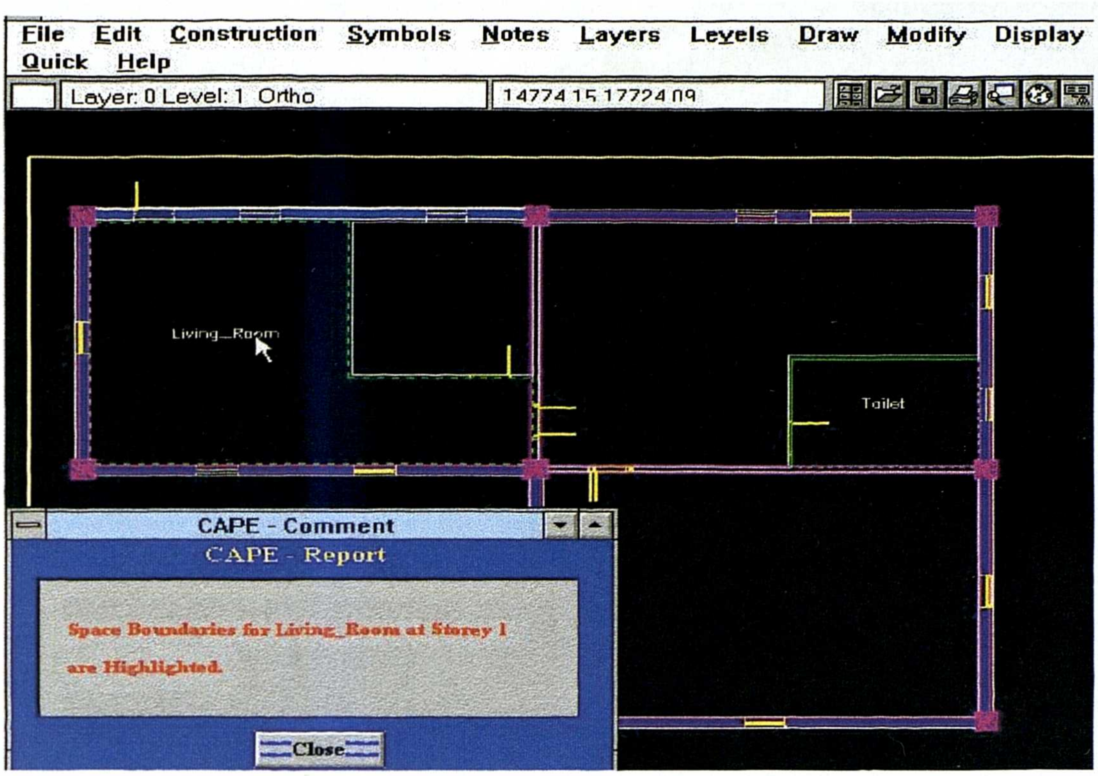


Figure 10.10: Highlighting the space boundaries of a living room

The surface finishes, which are attached to a defined space, can also be displayed. The floor surface finishes can be displayed by clicking on the space object or space name, while the wall surface finishes can be displayed by selecting any wall. This is shown in Figure 10.11. In a case where a wall is divided into two different spaces, all wall surface finishes attached to that wall will be displayed. Thus, this information enables users to associate a cost, a construction plan, etc. to a space according to their functionality. Users can therefore change a space attributes and examine their impact on cost, current requirements, construction, etc.

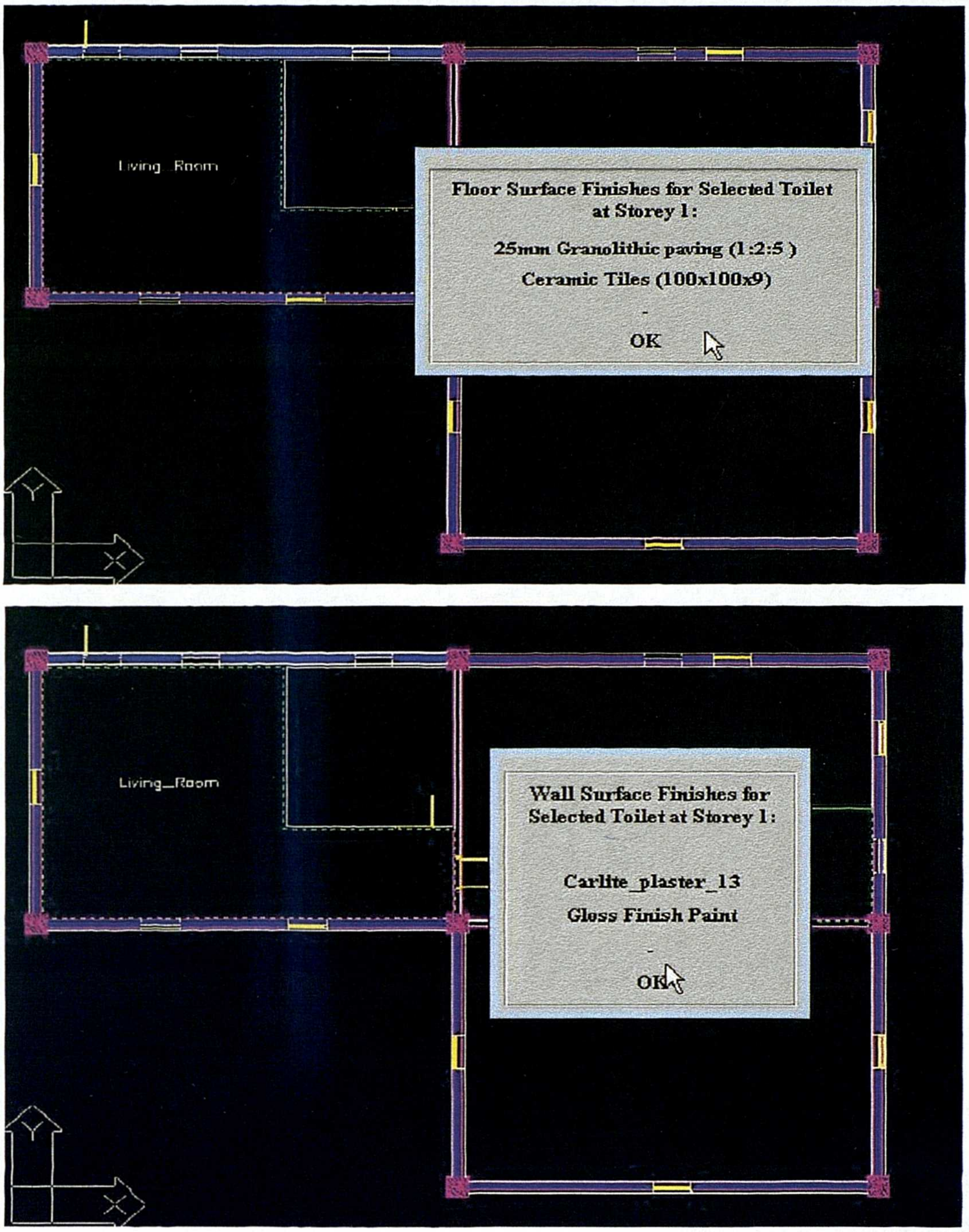


Figure 10.11: Displaying surface finishes for a selected space

10.3 Experimenting with the prototype

The aim of experimenting with the prototype is to validate the approach, applicability and usefulness of CAPE as the central core in an integrated construction environment (SPACE). The results of this experiment will lead to future recommendations on the approaches of implementing the systems which will be discussed in Chapter 11.

The process of experimenting with the prototype is divided into two parts:-

- The experimental approach
- Validation of the prototype systems

10.3.1 The experimental approach

In the world of research, the term "experiment" refers to controlled conditions in which one can extract and extrapolate results [cited from Kähkönen, 1993]. There are various approaches on experimenting with the prototype [Kähkönen, 1993; Pressman, 1992; DeMillo *et al.*, 1987]. Kähkönen [1993] highlighted two different approaches for experimenting the research. These approaches are:

1. An experiment in which data are extracted from the running or finished project.
2. An experiment in which based on laboratory research on data from fictional projects.

However, in this study the experiment was conducted using data from a fictitious completed building design where it is drawn in the AutoCAD-AEC™, i.e. a two-storey reinforced concrete building. The testing was performed in two stages. The first stage was to get a feedback from users/evaluators on the first part of CAPE, i.e. interpreting the design element while the second stage was to test the validity of the integration of CAPE with the other applications.

The specifications of the hypothetical building used for the testing of the prototype are as follows:-

- A reinforced-concrete two storey office building which has been divided into three main room space for each floor;
- Consists only square columns and rectangular beams;
- Consists of rectangular flat slabs and square pad footings;
- Flat slabs have been used for the roofing system;

Details of the structure frame of the building is shown in Figure 10.5. The numbers of the design elements are limited as mentioned earlier in the scope of the research in Chapter 1.

The testing was carry out by six people within the AIC research group including the developers of CONPLAN, EVALUATOR, INTESITE and CONVERT who have different experience background. These include civil engineer, structure engineer, construction planner and building engineer.

a) Interpreting the design elements

The demonstrations, which are highlighted earlier, have impressed the users/evaluators, especially on the topological relationship between the elements. The demonstration have achieved its aim, i.e. showing how CAPE manages to capture the topological relationship between two elements and other associated elements. However, the users/evaluators have suggested that if additional physical relationships are further defined between the design elements such as adjacent to, the results will be more accurate and reliable in determining the construction activities dependency. Other suggestion was made to add the clash detection functionality to the CAPE system to enable users to detect any unforeseen design related problems.

The method of identifying the topological relationships between two elements as soon as the element is defined has been commented on by the users/evaluators. As the number of elements increases, the time required for this process is also increased. This is mainly due to the process of matching the defined elements with all the previously drawn elements. The reason behind this is that in the wire-frame based CAD systems, the element cannot automatically detect the intersection between two elements. The worse case happened when an element is inserted into another element, such as beam in a column. Therefore, a long routine has to be carried out, i.e. by dividing the column into several 'virtual lines' in small intervals so that the column can behave like a solid element. The topological relationship is detected by finding any intersection between the lines of the two elements. Therefore, the suggestion is given to convert the wire-frame based CAD systems using a solid modelling technique.

Data from the building elements in the CAD systems are automatically interpreted as soon as the element is drawn and is then transferred to the project model. Many users/evaluators have made several suggestions and comments on the interpreted data and the method of transferring the interpreted data to the project model.

The method of automatically transferring the interpreted element and their data as soon as an element is drawn has been argued by some users/evaluators. If the element is somehow mistakenly drawn in CAD, its related information in the project model must be removed and updated. In this case, a special function needs to be developed in order to remove the element in the CAD system and also its associated objects in the project model.

The users/evaluators also suggested that the elements must be interpreted and transferred to the project model after the drawing has been finished in order to handle the problems stated above. Although this suggestion has its advantages to a certain extent, it will take considerable time to interpret each of the elements, i.e. especially if the building is too large. Also, any problems, which might be encountered during this long process, might be difficult to be detected and corrected.

b) Integration with other applications

CAPE is built to be the central core where the design information fed from the CAD system are stored, shared and manipulated by other construction applications. The integration between CAPE and other applications is done dynamically by interrogating the CAPE module as and when required. For example, when the design element is defined in the CAPE module, the SPECIFICATION module is automatically interrogated. When a construction plan needs to be generated, the CONPLAN module has to interrogate the CAPE module to access the design information. Thus, the way in which CAPE controls its data provides the flexibility in the integrated environment.

CAPE provides only the basic information required for performing a certain task. This is to ensure that only important and related information used by other applications are stored. For example, CAPE only provides the x, y, and z dimensions in which it could belong to width, length, height or depth of a certain element. The users/evaluators found that these information are very useful in providing the required information. When CAPE integrates with other applications, these three basic information can be used to provide other related information such as quantity, area, volumes, etc. which are required by other applications such as estimating, construction planning, etc. It was commented that the use of basic information such as those mentioned above could reduce the problem of data redundancies which might hinder the smooth development of an integrated system.

The capabilities of CAPE to integrate with CONVERT in virtual space provide a very useful tool to many users/evaluators. Three different applications have been developed by CONPLAN, INTESITE and EVALUATOR developers using CAPE and CONVERT to simulate their applications in virtual space. The construction simulation of CONPLAN shown in VR provides planners with a tool to evaluate the correctness and approachability of the plan. The simulation of site layout planning (INTESITE) allows the user to visualise the location and the situations of the facilities on the construction site. The fully/partly completed elements for the respective months, in a construction site, can be initially displayed and highlighted in the virtual space which could give the user the opportunity to enter the variation orders for the existing elements.

10.3.2 Validation of the prototype systems

The CAPE prototype provides many advantages to other SPACE developers such as CONPLAN, INTESITE, EVALUATOR and CONVERT. The approach used in the integration of CAPE with other application modules in the SPACE integrated environment provides many advantages, i.e. the ability of sharing and manipulating the design element's object at the early stage of their creation to the end of their life cycle.

The validation of the prototype has been carried out within the SPACE integrated environment. Some of the validation results are as follows:-

1. The demonstrations, which have been highlighted earlier have validated the importance of information which are stored in CAPE to different professionals.
2. The automatic generations of construction activities, bill of quantities, etc. by other researchers using the information stored in CAPE has validated the use of the design elements and their related information.
3. The capabilities of visualising and manipulating the design elements in the virtual space have proven that the information provided by CAPE can serve many applications as and when required.

4. The validated results from other researchers in the AIC research group who used CAPE data module as their main source of information have indirectly validate the results provided by CAPE.

10.4 Summary

CAPE was developed to provide the necessary information which is needed by other construction applications to perform their tasks. For example, the construction planning cannot be performed if the design information is not completed or the bill of quantities (BQ) cannot be determined if the dimensions of a design element is not provided. Therefore, the features of CAPE as the heart of an integrated construction environment (SPACE) which have been demonstrated can be further expanded to enrich the information base in the project model where more applications such as structural design application can be integrated.

The early consideration of handling the theoretical and the implementation issues while developing the framework and the information model has minimised the problems while integrating and supporting other applications. It has shown that such development has proven to be very effective to be the central resources for the integrated construction environments.

However, several suggestions and comments were highlighted by the users/evaluators in order to improve and enhance the general capabilities of the prototype. Among the suggestions and comments proposed are:-

1. Highlight the system's limitations while drawing the design elements in CAD.
2. Add a special application for removing and updating new elements in the CAD systems and project model.
3. Add additional physical relationships between the elements to provide higher accuracy while generating the construction plan.
4. Add the clash detection function to enable users to detect any unforeseen design related problems.
5. Improve the method of detecting the topological relationships between the elements using the solid modelling technique.
6. Allow the users/evaluators to interpret a group of elements and then transfer this to the project model.

Chapter 11

Summary and Conclusions

11.1 Introduction

The research in the field of integrated environments in the construction industry has been the focus of many researchers around the globe since early 1980's. Until now, the suitable "antidote" for the problems related to the implementation of such integrated environment have not been found yet. Due to this fact, this research has been motivated to find an approach where the problems, which contributed to the fragmentation in the construction industry, can be reduced/minimised.

Although previous studies have aimed towards producing a single integrated model to provide a common environment for all construction applications to interact with each other, some attempts have been directed towards the implementation issue, e.g. OSCON and WISPER. Moreover, the issue of management of information within the integrated environment has not yet been addressed fully. Therefore, the aim of this study is to overcome the above-mentioned problems which related to information management and information flow within and among the various data models in an integrated environment. This Chapter summaries the main stages of this study towards achieving its aims and objectives and outlines the main conclusions.

11.2 Summary of the research work

The traditional system of design and construction has led to a number of problems in the construction industry which has resulted in the decay of integration between the various professions and in misunderstanding of the role of each profession. The fragmentation occurs in both the horizontal and vertical directions. The horizontal direction is the communication between professionals involved in a particular phase of the project, whereas the vertical direction is the exchange of information between professionals involved in different phases of the project.

Due to these facts, the issue of managing the information flow between all the professionals in the project life cycle becomes a major need. Previous research has revealed that the theoretical and the implementation of handling this issue has been hindered by many constraints such as:-

- The limitations of the traditional approach to the design and construction process. This is where the design and construction are performed in sequential manner whereby a weak communication occurs between project participants due to the different views and perspectives of the information.

- The current contractual nature of the construction industry has led to additional implications for the sharing of information. Complex information flows, which are inherent in the industry, cannot easily be reduced by process improvement or by changes to contractual arrangement. This may cause a certain amount of reluctance from a designer, for example, having to share/exchange some of their information with other parties involved.

- The utilisation of sophisticated technologies in terms of hardware and software has created new problems. Although previous studies have proved that by using these technologies, the performance of the construction industry can be improved, the fragmentation in terms of hardware and software still occur. Most of the software packages are 'stand-alone' and tend to solve specific problems and business need only. This has created 'islands of automation' within the industry.

- The development and implementation of an integrated environment requires a common standard of data exchange and product models. However, the majority of the developed standards for data exchange and product models have been focused towards the theoretical developments in which the implementation issue for managing the information flow and information exchange/sharing within the integrated environment have not yet been addressed fully.

In order to overcome the above-mentioned constraints, a thorough literature review has been carried out to understand the issue of standards of data exchange and product models for managing the information flow in the construction process. The international standard, IGES and STEP have been developed as a mechanism for the representation and exchange of a computerised model of a product in a neutral form. Due to the complexity of the current information especially geometric information, a new standard has been developed namely the IFCs which aims towards providing intelligent objects for the construction industries that will dramatically increase the sharing of information throughout a building's life cycle, across disciplines and technical applications.

Although the developed standards seem to give some promises towards solving the integration problems, the number of interfaces are expected to be increased, i.e. one interface for each application. The introduction of a central project model, i.e. the product modelling approach, whereby all the construction applications can access and share the information can significantly reduce such problems. A number of previous developed product models for the integrated environments have been reviewed. It can be summarised that the development of a single product model should be addressed by:-

- Using an object-oriented data model where object-oriented technologies are the most effective and powerful approach for representing large bodies of complex engineering information;

- Emphasising the "abstraction level", i.e. the abstract of the building at the top level and the physical elements at the bottom level;

- Separating the domain into several object-oriented models, each describing the information needed to support well-defined activities;

- Considering the implementation issues, i.e. to the details of the domain in the lowest possible level;

- Considering the project life cycle in the project model starting from the inception to the demolition.

The above remarks will be incorporated in the development of an integrated environment whereby a developer's view needs to be created to clearly define its requirements and to present high level interactions between the various activities involved in the integration process. Structured Analysis and Design Technique (SADT) has been used to represent a proposed strategic framework for the Integrated Construction Environment (ICE), where design and other downstream applications can be integrated. The proposed approach is a representation of generic activities along with their relationships, which demonstrates how downstream applications can be integrated with the central core data models. The conceptual model is represented in three layers, starting from the Context Diagram down to level 1 process decomposition.

The above framework has been proposed for the Integrated Construction Environment (ICE) with the aim of co-ordinating the integration process between the various construction applications. The implementation of this framework SPACE, has led to the development of a modularised central core whereby each application has its own data module. The proposed structure consists of three main parts, i.e. the project model, software packages including interfaces with the project model, and external databases.

- The project model comprises of "building elements data module" and other application data and process modules. The "building elements data module" mainly describes the building's components and their behaviours. The extent and structure of this module depend on scope, context, and main objectives of the ICE, e.g. an environment for concrete framed buildings may have different building elements data structure of that of steel.

- The software packages represent the construction application packages such as CAD, construction planning, estimating, virtual reality, etc. Such application software packages can either be external, i.e. stand alone application packages, or internal, i.e. developed within the environment of the ICE. In either case, each application has its own user interface to manipulate the information, and especially developed two way communication channel to transfer information between the application and its related application data module at real time. These application packages are completely independent from the project model.

- The project model can retrieve external information from external databases as and when required by the various involved modules. This process can be carried out directly by the modules or shared by a number of applications modules, e.g. estimating and construction planning applications may need to share cost data which can be retrieved by any of these applications, say from on-line databases.

Such an environment is normally triggered off by feeding in the construction project specific information through a design package, for example, a CAD system where large amount of the project information can be extracted. Therefore, in order to extract the right information from the CAD system and populate it to the "object", the definition of the "object" has been defined. The main aim of having an "object" is to hold the project specific information where it can be shared or exchanged. However, due to the complexity and duplication of such information, it is not practically feasible to populate each "object" with what it requires over its life cycle. This will create data management problems, i.e. data control and maintenance, and data modelling difficulties. Therefore, to overcome this problem, the extracted data is divided into two groups, i.e. global data and specific data. Global data are attached to all elements as they are created while specific data are attached to a specific element.

In order to maintain the life of an object, which contains project specific information, a framework for object's life cycle has been proposed with the aim of understanding and formalising the behaviour of objects from creation to deletion. Four phases of object's life cycle have been highlighted. These are; create and amend, supplement object with data, use object and decommission object. Objects are populated with data at various phases of their life cycle. After instantiation, i.e. at the third phase of the life cycle, objects can either take a new status, i.e. when edited, or become mature objects. Objects at that stage can refer to their own data, to data that they have generated in other application data modules, or to common data. At the end of their life, they can either be deleted from the project model or become obsolete.

The above concept of an object and the structured framework of the object's life cycle have been mapped and represented in the form of conceptual models so that it can be identified and understood, and possibly agreed upon at the early stage of the development of an integrated environment. The EXPRESS-G modelling technique has been adopted in this study.

The developed "building elements data module" represents a framework for the presentation of design application, with the aim of establishing a computerised tool to assists both designers and construction professionals. It also aims for establishing a central core for the project model to facilitate the integration of design and construction. The relationship between the "building elements data module" and the other data modules has been discussed by stressing the impact of the "building elements data module" to support and share the project specific information in the ICE.

These data modules have been mapped into an object-oriented knowledge-based environment, KAPPA-PC^{TM}, as a central core in an integrated construction environment, SPACE. This has led to the development of a prototype, CAPE (Construction Application Protocols for data transfEr). The system's architecture of the prototype, CAPE comprises of three main parts; the Graphical Interface, the CAD System and the Object-Oriented Knowledge-Based System.

- The Graphical Interface allows the user to interact with the system. It consists of graphical packages, i.e. the AutoCAD-AEC^{TM} which is used to generate the design information while the Virtual Reality (VR) is used to visualise the design elements in virtual space.

- The CAD System is the main part of CAPE. It consists of three main parts, the Object Interpreter Engine (OIE), Graphical File Generator, and Space Analyser. The Object Interpreter Engine plays an important role in the CAD system whereby the design elements are extracted, interpreted, and transferred to the Object-Oriented Knowledge-Based System. The OIE also provides the information required for the Graphical File Generator whereby each created and interpreted object generates its graphical file in DXF format. The Space Analyser, on the other hand, analyses the building spaces whereby the co-ordination of the space boundary and the design elements, which bound the space, are captured.

- The Object-Oriented Knowledge-Based System stores the interpreted objects in a structured manner in order to serve other data modules. The Building Module is the main part where it stores the interpreted objects from the CAD system. It consists of several other data modules such as the Building Elements Data Module, the Building Space Data Module, etc. Each object is stored using a unique name created in the CAD system, which is mirrored to their graphical file name. The Building data Module provides data access to other data modules in SPACE environment whereby full data sharing and exchange take place.

The developed prototype CAPE has been experimented with. Two different demonstrations have been developed to show how the information captured and stored in CAPE can be utilised. The first demonstration is related to querying the design elements and the topological relationships, and displaying the design elements in VR. The second demonstration is related to the space analysis and their representation in the CAD system.

The implementation of CAPE in the integrated construction environment has provided an essential support for the integration of design and construction. CAPE not only provides most of the information required by other construction applications over the project life cycle, but it also provides a dynamic and an independent environment for all the graphical packages such as CAD and VR.

11.3 Main conclusions

The main conclusions of this study are listed below:-

1. The traditional approach in the project life cycle has hindered the smooth communication and co-ordination between the parties involved. This has led to a decay of integration between the various professions. Due to this fact, this study has focussed on the improvement of managing the information flow and how the information can be best shared and exchanged.

2. Design information is important to all stages in the project life cycle. Without the design information, other construction processes such as estimating, construction planning, etc. cannot be carried out. Therefore, the formalisation of the design information is essential in order to give a full support for other construction applications and the integration between the parties.

3. Recent studies have found that the construction industry can gain large benefits from the automation of its information flows and the creation of shared knowledge. The emergence of information technology (IT) has eased the various processes in the construction, such as design, planning, etc. but it intends only for solving

specific problems related to individual applications. Therefore, the automation and the integration between the various applications can increase data sharing, reducing time requirements and data error, accelerating communication among participants and improving completeness of information received by each team member.

4. The generic and strategic framework which has been proposed was found to be effective in supporting the development of a well-structured integrated construction environment. Three main issues have been proposed and highlighted, i.e. the data models, the applications and the project specific information. The data models are needed to satisfy the scope and the objectives of the environment's functionality. The applications, on the other hand, can be developed or linked with the central data models whereby the relevant information required by certain applications can be accessed and utilised from the data models. Finally, the project specific information are required for feeding the environment in order the applications can be implemented.

5. The modularised approach which has been introduced in this framework by separating the development of the applications individually and independently has proven to be essential to such development as it gives an excellent view on how the various parts of the ICE can be integrated. This enables individual experts to work separately and ease the data management of the environment. It also allows the testing procedure at any stage of the development which can highlight problems of inconsistencies and data duplication during the development stage.

6. The developed "Building Elements" data module has proved to support other data modules in the ICE whereby it is set as a central repository for other applications to share and integrate information.

7. The use of "representative object" technique to capture and group the primitive drawings of an element in CAD has provided the essential information required to support other construction applications. This has led to the improvement in the representation of the design elements, which originally consisted of many entities, by representing them as single entities/objects where they can be stored in the object-oriented knowledge-based system and can have their own graphical file (DXF).

8. The generated graphical file DXF is found to be flexible enough to support other graphical applications such as CAD and VR. The DXF file can be updated automatically as soon as an object is changed. For example, when a wall is inserted in a window, a DXF file for the new wall with an opening is created and update the previous file.

9. The use of Microsoft Windows™ environment with the facility of Dynamic Data Exchange (DDE) has given a real-time integration between the various applications.

10. The use of KAPPA-PC™ as an object-oriented development environment for rapid prototyping has proved to be appropriate. The prototype developed in this environment together with the use of VR has proved to show the power of CAPE as a central repository of the project specific information where any number of design

elements and the object relations can be visualised.

11. The use of KAPPA-PC™ for rapid prototyping has been limited by the allowable number of objects which can be created in KAPPA. This has limit the size of the hypothetical building used for the testing (see Chapter 10). The AutoCAD-AEC™ Version 3.1 also has limited the number of design elements in which some of the design elements such as beam and column have been customised to accommodate the prototype developed.

12. The prototype developed has achieved its aims and objectives, which have been stated in Chapter 1. This would pave the way towards improving the management of information flow and information sharing/exchange between all the participants in the construction industry. This would also provide an enhancement to the use of IT in integrating design and construction applications.

11.4 Recommendation for future work

The study has succeeded in achieving its aims and objectives towards facilitating the information flow and data exchange in an integrated construction environment. Some of the extended work are recommended to be further investigated and expanded are as follows:-

1. The industry standard, IFCs should be used as a method of information sharing in the construction industry. It will provide a "common language" for defining a

building project, a customisable industry-based object that encapsulate information about building elements as well as design, construction and management concepts.

2. The topological relationships between the design elements should be further investigated. Several other types of relationships and their correct definition would assist the construction planning application to accurately determine the dependencies between the construction activities.

3. The method for identifying the topological relationships should use the "solid modelling" technique instead of "wire-frame" technique which is largely used in the current CAD system. The "solid-modelling" provides a quick detection of any relationship between the design elements whereas the "wire-frame" requires several routines using the intersection method to detect any relationship.

4. The international standard data models provided by the STEP Application Protocols (APs), such as Building Construction Core Model (BCCM) should be used to facilitate the exchange of information worldwide. The BCCM for example, will enable the transformation of data between different discipline views in different project life cycle stages.

5. The strategic framework which has been developed can be extended to include more construction industry know-how (design and construction) and the product's specifications such as steel-framed building, bridge, etc. The modularised approach also can be improved by improving the method of sharing the information through the web-based approach.

6. The use of VR should be extended not only for visualisation purposes but also for assisting the user to assess, evaluate, exchange and update the project information as and when required.

11.5 Recommendations for the industry

The implementation of the integrated construction environment, SPACE has several benefits, which has been described earlier in section 6.7 of Chapter 6. It is seen to be a tool that would overcome current problems of integrating design and construction and could accommodate to serve the industry's needs. However, since it is only a prototype, several recommendations need to be further investigated. CAPE, which is part of SPACE and purposely designed to serve all the applications modules in the project model, requires further recommendations as follows:-

1. There must be a common standards of data exchange between the participants in the design and construction especially on the geometric data. The emergence of a new international standard, IFCs which aims to provide a method of information sharing in the construction industry.

2. The SPACE prototype provides industry with an integrated system which they can use for training and learning. Future requirement for integrated environments can be derived from using SPACE. Industrial feedback on the proper use of information is crucial for future developments.

3. It is also hoping that the new object-oriented CAD systems will be emerged in the industry. This will simplify and enhance the development process of interpreting the design elements whereby the interpreted objects could be stored directly in the CAD and if needed, it can be transferred to a central database.

4. The modularised approach which is used for the development of an integrated project database is recommended to be used as a standard approach in integrating different construction applications. It is found not only to be very effective and efficient but also provide the project model with the facility to be maintained and updated easily.

References

Abdalla. J. A. (1993), An Object-oriented Architecture and Concept for an Integrated Structural Engineering System, *CIVIL-COMP 93, Information Technology for Civil & Structural Engineering*, Topping. B. H. V. & Khan. A. I. (eds.), Civil-Comp Press, Edinburgh, August, pp. 147-155.

Ahuja. H. N., Dozzi. S. P. and Abourizk, S. M. (1994), *PROJECT MANAGEMENT: Techniques in Planning and Controlling Construction Projects*, second edition, John Wiley & Sons, inc., ISBN 0-471-59168-8.

Alagic. S. (1986), *Relational Database Technology*, Springer-Verlag, Texts and Monographs in Computer Science.

Alagic. S. (1989), *Object-oriented Database Programming*, Springer-Verlag, Texts and Monographs in Computer Science.

Alshawi. M. (1993), Time and Space Controlled 3D Model for the Construction Industry, *CIB-W65, Organisation and Management of Construction, 7th International Symposium*, September 15-22 1993, Port of Spain, Trinidad, pp. 201-206.

Alshawi. M. (1994a), A High Level and Real Time Integration Approach for CAD Systems, *Computing in Civil Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C System 94*, Khozeimeh. K. (ed.), June 20-22 1994, Washington DC, Vol.2, pp. 1779-1786.

Alshawi. M. (1994b), A Run Time Exchange of Elemental Information between CAD and Object Models: A Standard Interface, *The International Journal of Construction Information Technology,* Vol. 2, No. 2, pp. 37-52.

Alshawi. M. (1995), Object Definition and The Life Cycle of Objects, *CIB Workshop on Computers and Information in Construction, Modelling of Buildings Through Their Life Cycle*, M. Fischer, K. H. Law and B. Luiten (eds.), August 21-23 1995, Stanford, California, pp. 361-370

Alshawi. M. (1996), *Construction Integrated Environments*, Module 9 Workbook, A Module of the Masters in Construction IT IGDS Programme, University of Salford, UK.

Alshawi. M, Aouad. G., Child. T., Faraj. I. And Underwood. J. (1998), The Implementation of Integrated Environments using the Industry Foundation Classes Standards and Web Technologies, *The Life-Cycle of Construction IT Innovations: Technology Transfer from Research to Practice, Proceedings of the*

CIB W78, Björk. B. C. and Jägbeck. A (eds.), Royal Institute of Technology, Stockholm, Sweden, June 3 – 5, pp. 55-61.

Alshawi. M. and Che Wan Putra. CWF. (1994), Conceptual Modelling for 3D Simulation of Design and Construction, *Proceedings of the 3rd. Regional Conference on Computer Applications in Civil Engineering*, Nasly M. A (ed.), Kuala Lumpur, Malaysia, 2-4 August 1994, pp. 187-195.

Alshawi. M. and Che Wan Putra. CWF. (1995), Element Definition in an Integrated Design and Construction, *CIVIL-COMP 95, Developments in Computer Aided Design and Modelling for Civil Engineering*, B. H. V. Topping (ed.), 28-30 August 1995, Cambridge, England, Civil-Comp Press, pp. 87-92.

Alshawi. M. and Faraj. I. Z. (1995), Integrating CAD and Virtual Reality in Construction, *Proceedings of a Conference on Virtual Reality and Rapid Prototyping for Engineering*, University of Salford, January, pp.78-85.

Alshawi. M. and Underwood. J. (1996), Improving The Constructability Of A Design Solution Through An Integrated System, *The International Journal of Engineering, Construction and Architectural Management*, Vol. 3, No 1&2, pp 47-67.

Alshawi. M., Che Wan Putra. CWF. and Faraj. I. Z. (1997), A Framework for The Objects Life Cycle in Integrated Environments, *Microcomputers in Civil Engineering*, Vol.12, pp.339-351.

Aouad. G. (1992), *An Approach to System Analysis in the Construction Industry*, MSc IT in Construction Lecture Notes, Information Technology Institute, University of Salford, November 1992.

Aouad. G. (1996), Managing Construction Information Effectively Using Integrated Databases, *CIB W65, International Symposium for the Organisation and Management of Construction – Shaping Theory and Practice, Vol. 3 – Managing Construction Information*, Langford. D. A. and Retik. A. (eds.), E & FN Spon, pp. 14-22.

Aouad. G. and Price. A. D. F. (1993), Integrated Systems for the Construction Industry Possibilities, Problems and Limitations, *CIB-W65, Organisation and Management of Construction, 7th International Symposium*, September 15-22 1993, Port of Spain, Trinidad, pp. 257-268.

Aouad. G., Betts. M., Brandon. P., Brown. F., Child. T., Cooper. G., Ford. S., Kirkham. J., Oxman. R., Sarshar. M. and Young. B. (1994), *Integrated Databases for*

Design and Construction: Final Report, Internal Report, University of Salford, July.

Aouad. G., Kirkham. J., Brandon. P., Brown. F., Child. T., Cooper. G., Ford. S., Oxman. R. and Young. B. (1995), The Conceptual Modelling of Construction Management Information, *Automation in Construction*, Vol. 3, No. 4, pp. 267-282.

Aouad. G., Child. T., Brandon. P. and Sarshar. M. (1998), Linking Construction Information through VR using an Object Oriented Environment, *ARCOM, 14th Annual Conference*, University of Reading, UK, September 9-11, pp. 268 – 277.

Appleton. D. (1986), *Information Modelling Manual (IDEF1X)*, USAF Integrated Computer Manufacturing Program, Wright-Patterson AFB, O.H.

Ashford. J. L. (1989), *The Management of Quality in Construction*, E. &. F. N. Spon, London.

Atkin. B. L. (1989), *A Study of Information Technology in the Australian Construction Industry*, Unpublished Report, T. W. Crow Associates and Crow Maunsell Pty. Ltd. Management and Project Consultants, Sydney, Australia

Atkin. B. L. (1990), *Information Management of Construction Projects (Draft)*, Published by T. W. Crow Associates and Crow Maunsell Pty. Ltd. Management and Project Consultants, Sydney, Australia, ISBN: 0 7316 8420 6.

Atkin. B. L. (1995), Information Management of Construction Projects, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, London, ISBN: 0 419 203 70 2, Chapter 18, pp. 291-315.

Atre. S. (1988), *Data Base: Structured Technique for Design, Performance and Management*, John Wiley.

Augenbroe. G. (1992), Integrated Building Performance Evaluation in the Early Design Stages, *Journal of Building and Environment*, Vol. 27, No. 2, pp. 149-161.

Augenbroe. G. (1993), *COMBINE (Computer Models for the Building Industry in Europe)*, Final Report, Delft University of Technology, Netherland, July 1993.

Augenbroe. G. (1994a), Integrated Use of Building Design Tools: Results from the COMBINE Project, *In Tzonis. A. and White. I., Automation Based Creative Design*, Elsevier Science B.V.

Augenbroe. G. (1994b), Integrated Building Design Systems in Context of Product Data Technology, *Journal of Computing in Civil Engineering*, Vol. 8, No. 4, October 1994, pp. 420-435.

Augenbroe. G. (1995), Design Systems in a Computer Integrated Manufacturing (CIM) Context, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, pp194-212.

Augenbroe. G. and Rombouts. W. (1994), Extended Topology in Building Design Systems, *Computing in Civil Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C System 94*, Khozeimeh. K. (ed.), June 20-22 1994, Washington DC, Vol.2, pp. 1196-1203.

Australian STEP Demonstration Project (1995), *Improving Efficiency in the Construction Sector*, Version 1.6, August 26.

AutoDesk (1992a), *AutoCAD Release 12 - AutoCAD Customization Manual*, AutoDesk Inc., ISBN 2-88447-008-5.

AutoDesk (1992b), *AutoCAD Release 12 - AutoCAD Reference Manual*, AutoDesk Inc., ISBN 2-88447-003-4.

AutoDesk (1993), *Layer Naming Convention for CAD in Construction Industry*, Version 2, November 1993.

Betts. M., Cher. L, Mathur. K. and Ofori. G. (1991), Strategies for Construction Sector in the Information Technology Era, *Journal of Construction Management and Economics*, Vol.9, No. 6, December 1991, pp. 509-528.

Betts. M., Fischer. M. and Koskela. L. (1995), The Purpose and Definition of Integration, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, pp.3-18.

Betts. M., Mathur. K. and Ofori. G. (1989), *Information Technology and the Construction Industry of Singapore – A Framework for a Communication Network*, Report by Construction Management and Economics Research Unit, National University of Singapore, October.

Bichard. A.(1996), *The Bichard Column: The New AutoCAD ARX*, CADdesk Magazine, March 1996, Vol. 6, No. 3, pp. 14-15.

Birkbeck. D. (1991), *Tally Ho*, Technology Focus: Computers, Building, September, pp.69.

Björk. B. C. (1989), Basic Structure of a Proposed Building Product Model, *Computer-Aided Design*, Vol. 21, No. 2, pp. 71-77.

Björk. B. C. (1992a), A Unified Approach for Modelling Construction Information, *Journal of Building and Environment*, Vol. 27, No. 2, pp. 173-194.

Björk. B. C. (1992b), A Conceptual Model of Space, Space Boundaries and Enclosing Structures, *Automation in Construction*, Vol.1, No.3, pp. 193-214.

Björk. B. C. and Penttilä. H. (1989), A Scenario for the Development and Implementation of a Building Product Model Standard, *Proceedings of the Center for Integrated Facility Engineering*, Stanford, March 28-29.

Björk. B. C. and Wix. J. (1991), *An Introduction to STEP*, VTT and Wix McLelland, Ltd.

Brandon. P. and Betts. M. (1995), The Field of Integrated Construction Information: An Editorial Overview, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, pp. xxv - xxxi.

Brandon. P., Basden. A., Hamilton. I. and Stockley. J. (1988), *Expert Systems – The Strategic Planning of Construction Projects*, The RICS/ALVEY Research for Chartered Quantity Surveyors, RICS/University of Salford, ISBN 0 85406 334 X.

Branki. N. E. and Bridges. A. (1993), An Architecture for Cooperative Agents and Applications in Design, *Advanced Technologies, Fourth EuropIA International Conference on the Application of Artificial Intelligence, Robotics and Image Processing to Architecture, Building Engineering, Civil Engineering, Urban Design and Urban Planning*, Beheshti. M. R. and Zreik. K. (eds.), Delft, The Netherland, 21-24 June, pp. 221-230.

British Telecom (BT) (1995), *Construct IT Project – "Strawman" Final Report*, Report Draft 2.0, 24th January.

Broughton. T. (1990), Simultaneous Engineering in Aero Gas Turbine Design and Manufacture, *Proceedings of the first International Conference on Simultaneous Engineering*, December 4-5, London, pp. 25-36.

Building Project Information Committee (1987), *Co-ordinated Project Information and Common Arrangement of Work Sections*, 1st edition, June.

Canright. C. (1987), *EDI: the acronym that means business*, Systems/3X World, pp. 49-51.

Carty. G. J. (1995), Construction, *Journal of Construction Engineering and Management*, Vol.121, No. 3, September 1995, pp. 319-328.

CEN (1991), *Express Language Reference Manual*, CEN/CLC/AMT/WG STEP N48, Association Francaise de Normalisation, Paris, 160pp.

Che Wan Putra, C.W.F. (1991), *A Prototype Interpreter for Architectural CAD Drawing*, MSc Dissertation, Department of Surveying, University of Salford, United Kingdom, December.

Chen. P. (1976), *The Entity Relationship: Towards a Unified View of Data*, ACM Transactions on Database Systems, Vol.1, No.1, pp. 9-36.

Child. T., Tracey. A. and Aouad. G. (1996), *Open Systems for Construction (OSCON Interim Report)*, University of Salford, January 1996.

Choi. K. C. and Ibbs. C. W. (1989), Costs and Benefits of Computerization in Design and Construction, *ASCE Journal of Computing in Civil Engineering*, 3(1), pp.91-105.

CI/SfB (1976), *Construction Indexing Manual*, RIBA Publications Limited, 1976 Revision.

CIMSTEEL (Computer Integrated Manufacturing for Construction Steelwork) (1997), *http://www.leeds.ac.uk/civil/research/cae/**cimsteel**/*

Construction Industry Cost Effectiveness Project (1982), *Integrating Construction Resources and Technology into Engineering*, Technical Report B-1, New York, August 1982.

Construction Industry Research and Information Association (CIRIA) (1985), *Quality Assurance in Civil Engineering*, Report 109, CIRIA, London.

Cooper. G., Aouad.G., Brandon. P., Brown. F., Ford. S., Kirkham. J., Sarshar. M. and Young. B. (1993), Incorporating Alternative Perspectives in a Single Information Model. The ICON Project Analysis Method, *The Workshop on Models for Computer Integrated Construction*, Espoo, Finland, 5-9 October 1993.

Cornick. T. and Wix. J. (1988), Modelling of Building - Implications for Computer Systems, *Journal of Computer Integrated Construction*, Vol. 1, Issue 1, pp. 19-22. Wix McLelland Ltd., ISSN 0954-1128.

Crowley. A. J. (1996), *CIMsteel Integration Standards*, Source on World Wide Web at http://www.leeds.ac.uk/civil/research/cae/cis/cis.htm.

Cutting-Decelle. A.F. (1995), Standardization in Product and Process Data Modelling: The ISO STEP and MANDATE Standards' Contribution to the Integration of the Life Cycle of Buildings, *CIB Workshop on Computers and Information in Construction, Modelling of Buildings Through Their Life Cycle*, M. Fischer, K. H. Law and B. Luiten (eds.), August 21-23 1995, Stanford, California, pp. 196-210.

De Marco. T. (1978), *Structured Analysis and System Specification*, Prentice Hall Inc., Eaglewood Cliffs. N. J.

DeMillo, R. A., McCracken, W. M., Martin. R. J. and Passafiume. J. F. (1986), *Software testing and evaluation*, The Benjamin/Cummings Publishing Company Inc., California.

Department of Environment and Transport (DETR) (1996a), *Construct IT - Bridging the Gap Scoping Study for the Construction Industry Knowledge Base*, PE Consulting, June.

Department of Environment and Transport (DETR) (1996b), *Construct IT - Bridging the Gap: Implementation Plan 1996 – 2001*, on http://www.salford.ac.uk/survey/centex/cons_it/imp/contents.htm.

Department of Environmental and Transport (DETR) (1996c), *Feasibility of the Integrated Project Database*, IT Vision for the Industry at http://www.salford.ac.uk/survey/centex/ cons_it/reports/.

Dicesare. F. P. (1993), Re-engineering to Achieve a Concurrent Engineering Environment, *Journal of Design and Manufacturing*, Vol.3, pp. 75-89.

Dolan. D. F. (1979), *The British Construction Industry: an Introduction*, The Macmillan Press Ltd., ISBN 0-333-23552-5.

Dubois. A. M., Flynn. J., Verhoef. M. H. G. and Augenbroe. G. L. M. (1995), Conceptual Modelling Approaches in the COMBINE project, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 555-566, Balkema, Rotterdam, ISBN 90-5410-584-8.

Dym. C. and Levitt, R. (1991), *Knowledge-Based Systems in Engineering*, McGraw-Hill Inc., pp.182-183.

Eastman. C. M. (1978), The Representation of Design Problems and Maintenance of Their Structure, *Application of AI and PR to CAD, IFIPS Working Conference*, Latcombe (ed.), Grenoble, France, March, North-Holland, Amsterdam, The Netherland, pp.335-337.

Eastman. C. M. (1992), A Data Model Analysis of Modularity and Extensibility in Building Databases, *Journal of Building and Environment*, Vol. 27, No. 2, pp. 135 -148.

Eastman. C. M. (1994a), A Data Model for Design Knowledge, *Automation in Construction*, Vol. 3, pp. 135-147.

Eastman. C. M. (1994b), Integrated Product Models, *Computing in Civil Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C System 94*, Khozeimeh. K. (ed.), June 20-22, 1994, Washington DC, Vol.1, pp. 1188-1195.

Elmasri. R. and Navathe. S. B. (1989), *Fundamentals of Database Systems*, Benjamin/Cummings.

Enkovaara. E., Salmi. M. and Sarja. A. (1988), *RATAS Project - Computer Aided Design for Construction*, Building Book Ltd., Helsinki, 62p.

Euler. J. K. (1993), Construction Site Management Using Knowledge Based 3D Animated CAD Models, *A/E/C SYSTEMS Computer Solution*, May-June 1993, pp. 7-14.

Evbuomwan. N. F. O. and Anumba. J. C. (1995), Concurrent Life Cycle Design and Construction, *Developments in Computer Aided Design and Modelling for Civil Engineering*, B.H.V. Topping (ed.), CIVIL-COMP Press, Edinburgh, UK, pp. 93-102.

Evbuomwan. N. F. O. and Sivaloganathan. S. (1995), Views on the Concurrent Engineering Paradigm in Civil and Structural Engineering, *CIVIL-COMP 95, Developments in Computer Aided Design and Modelling for Civil Engineering*, Topping. B. H. V.(ed.), pp.107-113.

Everest. G. C. (1986), *DATABASE MANAGEMENT: Objectives, System Functions, and Administration*, McGraw-Hill Book Company, ISBN 0-07-019781-4.

Ewen. I. and Alshawi. M. (1993), An Object Oriented Approach for Architectural CAD Drawings, *CIVIL-COMP 93, The Application of Artificial Intelligence to Civil and Structural Engineering*, B. H. V. Topping (ed.), August 1993, Edinburgh, Civil-Comp Press, pp. 167-174.

Faraj. I. Z. (1994), *Verification and Machinability of Manufacturing Features in a Product Modelling Environment*, PhD Thesis, Department of Mechanical Engineering, University of Leeds, UK.

Faraj. I. Z. and Alshawi. M. (1996), A Modularized Approach to the Integrated Environment, *Construction Information Highway: Proceedings of CIB Workshop – W78*, Bled, Slovenia, June 10th – 12th, Turk. Z. (ed.), pp. 179-198.

Fereshetian. N. and Eastman. C. (1991), A Comparison of Information Models for Product Design, *paper presented at the CIB seminar, The Computer Integrated Future.*

Ferry. D. J. and Brandon. P. S. (1991), *Cost Planning of Buildings*, 6th Edition, Oxford BSP Professional Books.

Fischer. M. (1989), Design Construction Integration Through Constructibility Design Rules for the Preliminary Design of Reinforced Concrete Structures, *Proceedings of CSCE/CPCA Structural Concrete Conferences*, March 20-21 1989, Montreal, Canada, pp. 333-346.

Fischer. M. (1991), Reasoning about Constructibility: Representing Construction Knowledge and Project Data, *Proceedings of 2nd International Conference on* *the Application of Artificial Intelligent to Civil and Structural Engineering*, Topping. B. H. V. (ed.), Civil-Comp Press, September, pp.105-112.

Fischer. M. and Froese. T. (1992), Integration Through Standard Project Models, *Joint CIB Workshop on Computers and Information in Construction*, Vanier. D. J. and Thomas. J. R. (eds.), Montreal, Canada, May, pp. 189-205.

Fischer. M., Froese. T. and Phan. D. (1994), How Do Integration and Data Models Add Value to a Project?, *Computing in Civil Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C System 94*, Khozeimeh. K. (ed.), June 20-22 1994, Washington DC, Vol.1, pp. 992-997.

Fischer. M., Betts. M., Hannus. M., Yamazaki. Y. and Laitinen. J. (1993), Goals, Dimensions and Approaches for Computer Integrated Construction, *Proceedings of the First International Conference of the Management of Information Technology for Construction*, Mathur. K. S., Betts. M. P. and Tham. K. W. (eds.), National University of Singapore, August, pp. 421-433.

Fischer. M., Froese. T. and Phan. D. (1994), How Do Integration and Data Models Add Value to a Project?, *ASCE Computing in Civil Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C Systems '94*, Khozeimeh. K (ed.), pp. 992-997.

Forster. G. (1986), *Building Organisations and Procedures*, 2nd Edition, Longman Scientific and Technical, Essex, England, ISBN 0-582-41373-7.

Froese. T. (1992), *Integrated Computer-Aided Project Management Through Standard Object-Oriented Models*, PhD Thesis, Department of Civil Engineering, Stanford University, USA.

Froese. T. (1993), Project Modeling and Data Standards for AEC, *Computing in Civil and Building Engineering, ASCE Proceedings of the 5th International Conference*, L. F. Cohn (ed.), Anaheim, California, June 7-9, 1993, pp. 1730-1737.

Froese. T. (1994a), *A Summary of: Workshop on the Standardisation of Information Structures in the Construction Industry*, The Royal Institute of Technology, Stockholm, Sweden, April 6-7, 1994 (Distributed on the step-aec@NIST.GOV Internet e-mail distribution list, April 27, 1994).

Froese. T. (1994b), Information Standard in the AEC Industry, *Canadian Civil Engineer*, Vol. 11, No. 6, September 1994, pp. 3-5.

Froese. T. (1995a), Core Process Models and Application Areas in Construction, *CIB Workshop on Computers and Information in Construction, Modelling of Buildings*

Through Their Life Cycle, M. Fischer, K. H. Law and B. Luiten (eds.), August 21-23, Stanford 1995, California, pp. 427-436.

Froese. T. (1995b), Models of Construction Process Information, *Computing in Civil Engineering, ASCE Proceedings of the 2nd. Congress Held in Conjunction with A/E/C Systems '95*, J. P. Mohsen (ed.), Atlanta, GA, June 5-8, Vol. 1, pp. 5-12.

Froese. T. and Paulson. B. C., Jr. (1994), OPIS: An Object Model- Based Project Information System, *Microcomputers in Civil Engineering*, Vol.9, No.1, pp.13-28.

Gardiner. P. D. and Simmons. J. E. L. (1992), Analysis of Conflict and Change in Construction Projects, *Journal of Construction Management and Economics*, Vol. 10, No. 6, November 1992, pp. 459-478.

Geilingh. W. F. (1989), *General AEC Reference Model*, ISO TC 184/SC4/WG1 Document N329, ISO.

Geilingh. W. F. (1990), Computer Integrated Construction: A Major STEP Forward, *Journal of Computer Integrated Construction 1*, Wix McLelland Ltd.

Gielingh. W. (1993), *PISA Framework,* in PISA Workshop Report, Paris, January 27-28.

Gielingh. W. and Suhm. A. K. (1993), *IMPPAXCT Reference Model*, Project 2165, IMMPACT, 1, Springer Verlag.

Goldscmidt. E. and Navon. R. (1996), Automated Real-Time Manpower Productivity Control, *CIB W65 International Symposium for the Organisation and Management Construction – Shaping Theory and Practice, Vol. 3 – Managing Construction Information*, Langford. D. A. & Retik. A. (eds.), E & FN Spon, pp.199-205.

Grabowski. H., Anderl. R. and Pratt. M. J. (1991), *Advanced Modelling for CAD/CAM Systems*, Research Report ESPRIT, Project 322: CAD Interfaces (CAD*1), Springer-Verlag, Vol. 7, ISBN 3-540-53943-3.

Greening. R. (1995), Product Models and Data Exchange, *Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering*, P. J. Pahl and H. Werner (eds.), 12-15 July 1995, Berlin, Germany, Balkema, Rotterdam, pp. 299-303, ISBN 90-5410-556-9.

Greening. R. and Edwards. M. (1995), ATLAS implementation scenario, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 467-472, Balkema, Rotterdam, ISBN 90-5410-584-8.

Harvey, R. C. and Ashworth. A. (1993), *The Construction Industry in Britain*, Butterworth-Heinemann Ltd., London, ISBN 0-7506-0359-X

Hassan. Z. (1997), *CONPLAN: Construction Planning and Buildability Evaluation in an Integrated and Intelligence Construction Environment*, PhD Thesis, Department of Surveying, University of Salford, UK.

Hendrickson. C. and Maher. M. L. (1989), Issues in Computer Based Design/Construction Integration, *Excellence in the Constructed Project, ASCE Proceedings of Construction Congress I*, Bard. R. J. (ed.), pp. 129-136.

Hon. S. L., Gairns. D. A. and Wilson. O. D. (1989), Buildability: A Review of Research and Practice, *Australian Institute of Building Papers*, Vol.3, pp. 101-118.

Howard . H. C., Levitt. R. E., Paulson. B. C., Pohl. J. G. and Tatum. C. B. (1989), Computer Integration: Reducing Fragmentation in AEC Industry, *ASCE Journal of Computing in Civil Engineering*, Vol.3, No. 1, January, pp. 18-32.

Howards. H. C. (1989), Prototype Integrated Environment for AEC Data, *ASCE Proceedings of the 6th Conference of Computing in Civil Engineering-Computers in Engineering Practice*, Barnwell. T. O. Jr. ed.), pp. 458-465.

Howell. I. (1996), The Need for Interoperability in the Construction Industry, *INCIT – International Construction Information Technology 96 Proceedings*, The Institution of Engineers, Australia, Sydney, 18-19 April, pp. 43-47.

Ibbs. C. W. and Choi. K. C. (1989*), Cost Effectiveness of Computerization in Design and Construction*, Construction Engineering and Management, Technical Report No. 13, University of California at Berkeley, February.

Ikeda. M., Sekihara. Y. and Itoh. N. (1996), Construction Planning System for High-Rise Buildings using an Object-Based Model, *Construction of the information Highway: Proceedings of the CIB Workshop W78*, Turk. Z. (ed.), Bled. Slovenia, June 10-12, pp.283-296.

Industry Alliance for Interoperability (IAI) (1995), *Industry Foundation Classes (IFC)*, Autodesk Expo, A/E/C System's 95, Stanford, June 1995.

Industry Alliance for Interoperability (IAI) (1996), *Industry Foundation Classes - IFC Project Model Specifications*, Draft 4-10 May 1996, Version 0.94.

Iosifidis. P., Tah. J. H. M. and Howes. R. (1995), *An Advanced Object-Oriented Architecture for Information Exchange through Shared Objects*, Engineering Computing Newsletter, Issue 59, November, pp. 4-5.

IRMA (1993), *Transcripts of IRMA-tica 93, An International E-Mail Conference on the IRMA Model: An Information Reference Model for AEC*, Available for anonymous ftp or gopher file transfer from irma.hs.jhu.edu.

ISO (1991), *ISO: Industrial Automation Systems Product Data Representation and Exchange. STEP part 1: Overview and Fundamental Principles*, October 1991, TC184/SC4/MAG/N43, Version 8.

ISO 6707-1 (1989), *Building and Civil Engineering – Vocabulary*, Part 1, General Terms.

ISO EXPRESS Committee (1990), *The Express Language*, July 1990, ISO TC184/SC4 N64.

ISO TC184/SC4/WG3 (1994), *Building Construction Core Model*, Document N341, October 1994.

Ito. K., Law. K. H. and Levitt. R. E. (1990), PMAPM: An Object-Oriented Project Model for A/E/C Process with Multiple Views, *2nd CIB W78 + W74 Seminar on Computer Integrated Construction*, Tokyo, Japan, September 17-19, pp.75-79.

Jacobus Technology Inc. (1994), *Jspace Object Engine*, Jacobus Technology Inc. 7901, Beechcraft Avenue, Gaithersburg, MD 20579.

Jegaraj. T. A. (1983), Coordination as an Effective Tool for Project Management, *Seminar on Project Management in Malaysia*, 29th April 1983, pp. 1-17.

Jennings. I. C. and Kenley. R (1996), Integrated Complexity, *CIB W65 International Symposium for the Organisation and Management Construction – Shaping Theory and Practice, Vol. 2 – Managing the Construction Project and Managing Risk*, Langford. D. A. & Retik. A. (eds.), E & FN Spon, pp. 339-351.

Jones. B. and Riley. M. J. (1993), Concurrent Construction Integrated Design, CIVIL-COMP 93, *Proceedings of Knowledge Based System for Civil and Structural Engineering*, B. H. V. Topping (ed.), August 1993, Edinburgh, Civil-Comp Press, pp. 53-60.

Jones. F. H. and Martin. L. (1991), *The AutoCAD Database Book: Accessing and Managing CAD Drawing*, 4th Edition, Chapel Hill, N.C.: Ventana Press.

Jordani. D. A. (1993), *Integration: Opportunity of Change*, A/E/C SYSTEMS Computer Solution, May-June 1993, pp. 46-50.

Junge. R. and Liebich. T. (1995), Product Modelling for Applications: Models for Next Generation CAAD, *Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering*, P. J. Pahl and H. Werner (eds.), 12-15 July 1995, Berlin, Germany, Balkema, Rotterdam, pp. 309-315, ISBN 90-5410-556-9.

Kähkönen, K. (1993), *Modelling Activities Dependencies for Building Construction Project Scheduling*, VTT Publications 153, Espoo, Finland.

Kahn. H. J. (1995), STEP Methodology - An Overview, *Information Technology Awareness in Engineering, Object Technology and Its Application in Engineering*, J. A. Powell (ed.), March 1995, Salford, UK, pp. 66-75.

Kangari. R. and Sadri. S. (1996), Building Construction Primary Task Models, *Construction of the Information Highway: Proceedings of CIB Workshop W78*, Turk. Z. (ed.), Bled, Slovenia, June 10-12, pp. 319-333.

Kartam. N. A. (1994), ISICAD: Interactive System for Integrating CAD and Computer-Based Construction Systems, *Microcomputers in Civil Engineering*, Vol. 9, No. 1, pp. 41-51.

Katranushkov. P. (1995), COMBI: Integrated product model, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 511-520, Balkema, Rotterdam, ISBN 90-5410-584-8.

Keijer. U. (1992), Future Organisation of the Building Process. *Swedish National Report to the CIB W82, Symposium Construction Beyond 2000*, June 15-18 1992, Espoo, Finland, Swedish Council for Building Research, Stockholm, 100pp.

Koskela. L. (1995), On Foundations of Construction Process Modeling, *CIB Workshop on Computers and Information in Construction, Modelling of Buildings Through Their Life Cycle*, M. Fischer, K. H. Law and B. Luiten (eds.), August 21-23 1995, Stanford, California, pp. 503-514.

Levitt. R. E., Axworthy. A. and Kattajamäki. M. (1991), Automating Engineering Design With Design + +, *Nikkei AI Journal,* Tokyo, Japan, (Winter), pp. 118-127.

Luiten. B. et al. (1993), An Information Reference Model for Architecture, Engineering and Construction, *Proceedings of the First International Conference on the Management of Information Technology for Construction*, Mathur. K. S. et al. (eds.), August 1993, World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 391-406.

Marca. D. A. and McGawn. C. L. (1988), *SADT: Structured Analysis and Design Technique*, McGraw-Hill Book Company, New York, ISBN 0070402353.

Martin. J. (1993), *Principle of Object-Oriented Analysis and Design*, PTR Prentice-Hall Inc., ISBN 0-13-720871-5.

McIntyre. C., Parfitt, K. and Syal. M. (1994), CAD-Based "Intelligent" Drawings: An Automated Quality Control Approach to Land Development, *Computing in Civil*

Engineering, Proceedings of the First Congress Held in Conjunction with A/E/C System 94, Khozeimeh. K. (ed.), June 20-22 1994, Washington DC, Vol.2, pp.547-554.

McKay. A. and Bloor. M. S. (1991), The Role of Product Model in Effective CADCAM, *Effective CADCAM'91: Taking Advantage of Engineering Technologies*, pp. 113-119.

McKay. A. and Bloor. M. S. (1992), *Product Modelling Seminar*, Mechanical Engineering Department, University of Leeds, UK, July.

Minemasa. K. (1995), Proposal for Integrated Production Data System Structure for Creation and Delivery of Production Data in Order to Improve Productivity, *Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering*, P. J. Pahl and H. Werner (eds.), 12-15 July 1995, Berlin, Germany, Balkema, Rotterdam, pp. 429-434, ISBN 90-5410-556-9.

Miyatake. Y. and Kangari, R. (1993), Experiencing Computer Integrated Construction, *Journal of Construction Engineering and Management*, Vol.119, No.2, pp. 307-322.

Munns. A. K., Li. Y. B. and Lookman. A. A. (1995), Strategy for Computer Integrated Design Construction and Technology, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 51-57, Balkema, Rotterdam, ISBN 90-5410-584-8.

Navon. R. (1995), COCSY I: CAM Oriented CAD System, *ASCE Journal of Computing in Civil Engineering*, Vol. 9, No. 4, pp.236-243.

Nevins. D. P. and Zabilski. R. J. (1989), An integrated System for Fossil Power Plant Construction and Retrofit Projects, *1st International Conference on Fossil Plant Construction*, Armor. A. (ed.), Electric Power Research Institute, Palo Alto, California.

Nijssen. G. M. and Halpin. T. A. (1989), *Conceptual Schema and Relational Database Design, A Fact Oriented Approach*, Prentice Hall, London, 342pp.

O'Brien. M. J. and Al-Soufi. A. (1993), Electronic Data Interchange and the Structure of the UK Construction Industry, *Journal of Construction Management and Economics*, Vol. 11, No. 6, Nov.1993, pp. 443-453.

O'Connor. T. J. and Tucker. R. L. (1986), Improving Industrial Project Constructibility, *Journal of Construction Engineering and Management*, Vol. 112, No. 1, March 1986, pp. 69-82.

Palmer. M., Gilbert, M. and Anderson, J. (1993), *Guidelines for the Development and Approval of STEP Application Protocols*, Version 1.1, National Institute of Standards and Technology (NIST)..

Pasquill. K. G. (1988), Practical Uses of IGES, *Computer Aided Design*, Vol. 20, No. 6, July/August 1988, pp. 358-359.

Pearson. M. P. (1989), Advantages of 3D Design in Civil Engineering, *Computer in Engineering Practice, Computing in Civil Engineering, ASCE Proceedings of the 6th Conference*, Barnwell. T. O. (ed.), pp. 805-810.

Pisano. G. and Wheelwright.S. (1995), *The New Logic of High-Tech R&D*, Harvard Business Review, Sept.-Oct., pp. 93-105.

Platt. D. G. and Blockley. D. I. (1994), Process Modelling in Civil Engineering Design, *The International Journal of Construction Information Technology*, Vol. 2, No. 1, pp. 33-44.

Polytechnic of Central London (1984), *Problems of Buildability: Their Prevalence and Solutions*, Final Report to the Construction Industry Research Information Association (CIRIA), Department of Building, RIBA Research Group, London.

Poyet. P. (1995), ATLAS integration tools, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 479-488, Balkema, Rotterdam, ISBN 90-5410-584-8.

Pressman. R. G. (1992), *Software Engineering: A Practitioner's Approach*, McGraw-Hill Inc.

Project Management Institute (PMI) (1996), *A Guide to the Project Management Body of Knowledge*, on http://www.eniac.com/whitepap/project.htm.

PSM Report (1995), *Knowledge Base International*, Vol. 4, No. 12, December 1995, pp. 16-17.

Ramscar. M. (1994), Static Models and Dynamic Designs - An Empirical Impasse Vs an Inductive Solution, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 69-76, Balkema, Rotterdam, ISBN 90-5410-584-8.

Rasdorf. W. J. and Abudayyeh. O. (1992), NIAM Conceptual Database Design in Construction Management, *Journal of Computing in Civil Engineering*, Vol. 6, No. 1, January 1992, pp. 41-62.

Ratcliffe. J. (1985), *Project Management for Property Development*, Occasional Papers, Department of Estate Management, Polytechnic of South Bank, London.

Reed. K. A (1988), Product Modelling of Buildings for Data Exchange Standards: from IGES to PDES/STEP and Beyond, *Proceedings of CIB W74 + W78 Seminar on Conceptual Modelling of Buildings*, Lund, Sweden, 24-28 October 1988, pp.157-164.

Reinschmidt. K. F. and Zabilski. R. J. (1989), Applications of Computer Graphics in Construction, *Proceedings of Construction Congress I - Excellence in the Constructed Project*, ASCE, New York, pp. 137-142.

Reinshmidt. K. F., Griffis. F. H. and Bronner. P.L. (1991), Integration of Engineering, Design and Construction, *Journal of Construction Engineering and Management*, Vol. 117, No. 4, December, pp. 756-772.

Riley. M. J. and Sabet. H. H. R. (1994), Building Product Model, A First Brick in Computer Integrated Construction, *ASCE Computing in Civil Engineering,* *Proceedings of the First Congress Held in Conjunction with A/E/C Systems '94*, Khozeimeh. K (ed.), pp. 767-777.

Ross. D. T. (1977), *Structured Analysis: A Language for Communicating Ideas*, IEEE Transaction on Software Engineering, January, SE-3(1), pp. 16-34.

Ross. D. T. and Schoman Jr., J. K. (1977), *Structured Analysis for Requirements Definition*, IEEE Transaction on Software Engineering, January, SE-3(1), pp. 6-15.

Royal Institute of British Architects (RIBA) (1980), *RIBA Handbook of Architectural Practice and Management*, 4th Revised Edition, RIBA Publication Limited, London.

Rusell. A. and Froese. T. (1995), Design Challenges for Contractors' Integrated Management Systems, *ASCE Computing in Civil Engineering, Proceedings of the Second Congress Held in Conjunction with A/E/C Systems '95*, Mohsen. J. P. (ed.), Atlanta, GA. June 5-8, Vol. I & II, pp. 486-493.

Saarnivaara. V. P. (1990), Information Technology as a Tool for Industrialised Construction Systems and Processes, *2nd Finish-French Colloquium for Information Technology in Construction*, VTT Symposium 119, pp. 10-30.

Sadri. S. L. and Kangari. R. (1993), Integrated Information Systems in Construction: Strategic Technology, *CIB-W65, Organisation and Management of Construction, 7th International Symposium*, September 15-22 1993, Port of Spain, Trinidad, pp. 193-199.

Sanders. J. (1988), *EDI: Electronic Data Interchange for the UK Construction Industry*, Construction Computing, July 16-18.

Sanvido. V. E. (1992a), Linking Levels of Abstraction of a Building Design, *Building and Environment*, Vol. 27, No. 2, pp. 195-208.

Sanvido. V. E. (1992b), Penn. State Computer Integrated Construction Research Program, *Joint International Workshop on Computers Integrated Construction and Computers and Building Standards*, D. J. Vanier and J. R. Thomas (eds.), May 12-14, 1992, Montreal, Canada, pp.394-400.

Sanvido. V., Khayyal. S. A., Guvenis. M. M., Norton. K. J., Hetrick. M. S., Muallem. M. A., Chung. E. K., Medeiros. D. J., Kumara. S. R. T. and Ham. I (1990), *An Integrated Building Process Model*, Technical Report No. 1, Computer Integrated Construction Research Program, The Pennsylvania State University, January.

Schenck. D. (1989), *Express Information Modelling Language Reference Manual*, N386, ISO TC 184/SC4/WG1.

Schenck. D. and Wilson. P. (1994), *Information Modelling the EXPRESS way*, Oxford University Press.

Scherer. R.J. (1995), EU-project COMBI-Objectives and overview, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 503-510, Balkema, Rotterdam, ISBN 90-5410-584-8.

Schimming. B. B. (1993), *The Coming Competition*, Forum Civil Engineering, Vol. 53, p.6.

Shrensker. W. L. (1990), *CIM: A Working Definition*, SME Blue Book Series, Dearborn, MI, Society of Manufacturing Engineers.

Sibley. E. H. (1974), Data Management Systems - User Requirements, *Database Management Systems, 1973 SHARE Working Conference*, D. Jardine (ed.), North Holland Press, N. Y., pp. 83-104.

Simons. K. L., Thornberry. H. L. and Wickard. D. A. (1988), Simulation System for Construction Planning and Scheduling, *Joint ASME/IEEE Power Generation Conference*, ASME, New York.

Smart. D. A. (1987), The Importance of a CAD/BQ Estimating Link for the 1990's, *Transaction of the Building Cost Research Conference on Building Cost Modelling and Computer*, Brandon. P. S. (ed.), E & FN Spon, London, pp.119-126.

Smith. B. (1986), International efforts in product data exchange, *(In Proc.), MICAD86*, pp. 751-785, Hermes, Paris.

Standard Method of Measurement of Building Works 7th edition (SMM7) (1991), *Wessex SMM7 Building Price Book*, Wessex Electronic Publishing Limited, 1st edition, August.

Sulaiman. M. J. (1997), *INTESITE: An Intelligent Site Layout Planning within an Object-oriented Integrated Construction Environment*, PhD Thesis, Department of Surveying, University of Salford, UK.

Syan. C. S. and Menon. U. (1994), *CONCURRENT ENGINEERING: Concepts, Implementation and Practice*, Chapman & Hall, London, UK, ISBN 0-412-58130-2.

Tah. J. H. M., Howes. R. and Iosifidis. P. (1994), Capturing Semantic Data in CAD for Construction Project Planning, *Proceedings of ECPPM'94 – The First European Conference on Product & Process Modelling in The Building Industry*, Dresden, Germany, 5-7 October, pp. 287-293.

Task Committee on Application of Small Computers in Construction of the Construction Division (1985), Application of Small Computers in Construction, *Journal of Construction Engineering and Management*, Vol.111, No.3, September, pp.173-189.

Teicholz. P. (1989), Integration of Microcomputer Applications - Current and Future Approaches, *Excellence in the Constructed Project, ASCE Proceedings of Construction Congress I*, Bard. R. J. (ed.), pp. 147-158.

Teicholz. P. and Fischer. M. (1993), Strategy for Computer Integrated Construction Technology, *Journal of Construction Engineering and Management*, Vol. 120, No. 1, pp. 117-131.

Terk. M. and Fenves. S. J. (1995), Using Contract Nets in Integrated Design Environment, *ASCE Computing in Civil Engineering, Proceedings of the Second Congress Held in Conjunction with A/E/C Systems '95*, Mohsen. J. P. (ed.), Atlanta, GA. June 5-8, Vol. I & II, pp. 1098-1105.

The Building Centre Trust (1991), *Building IT 2000*, May.

The National Building Agency (NBA) (1972), *Project Co-ordination: The Pre-contract Planning and Control of Building Works Programmes*, NBA, London.

Tolman. F. and Poyet. P. (1995), The ATLAS models, *The 1st European Conference on Product and Process Modelling in the Building Industry*, R. J. Scherer (ed.), 5-7 October 1994, Dresden, Germany, pp. 473-477, Balkema, Rotterdam, ISBN 90-5410-584-8.

Tolman. F., Bakkeren. W. and Böhms. M. (1994), *Atlas LSE Project Type Model*, ESPRIT Project 7280 - ATLAS/WP1/Task 1500 Document D 106-Ic, April.

Toms. P. (1992), *Categorisation of Design Information in Terms of Design Process and Product Specifications*, A Standard for Computer Representation of Specification.

Tucker. S. N. (1996), *Personal Communication. Project Leader, Construction Process Re-engineering*, CSIRO Division of Building, Construction and Engineering, Melbourne, Australia.

Tucker. S. N. and Mohamed. S. (1996), Introducing Information technology in Construction: Pains and Gains, *CIB W65 International Symposium for the Organisation and Management Construction – Shaping Theory and Practice, Vol. 3 – Managing Construction Information*, Langford. D. A. & Retik. A. (eds.), E & FN Spon, pp. 348-356.

Turner. J. A. (1988), A Systems Approach to the Conceptual Modelling of Buildings, *Proceedings of CIB W74 + W78 Seminar on Conceptual Modelling of Buildings*, Lund, Sweden, 24-28 October 1988, pp.179-187.

Turner. J. A. (1990), *The Conceptual Modeling of a Building*, Computer Integrated Construction (Newsletter), Vol.1, No.1, Wix-McLelland publications, Thatcham, UK.

Tzanev. D. (1995), Integration of Product and Process Models in the Object-Oriented CAD Systems, *Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering*, P. J. Pahl and H. Werner (eds.), 12-15 July 1995, Berlin, Germany, Balkema, Rotterdam, pp. 353-357, ISBN 90-5410-556-9.

Underwood. J. and Alshawi. M. (1996), *Project Estimates and Monthly Interim Valuation Certificates Generation in an Integrated Construction Environment*, Intermediate Report, University of Salford, May.

Underwood. J. and Alshawi. M. (1997), Data and Process Models for the Integration of Estimating and Valuation, *Microcomputers in Civil Engineering*, Vol. 12, pp. 393-405.

University of Minnesota (1997), *CE5703-Project Management*, on http://www.ce. umn.edu/courses/ce5703/MMI/MMI3.html.

Vajpayee. S. K. (1995), *Principles of Computer Integrated Manufacturing*, Prentice-Hall Inc., ISBN 0-02-422241-0.

Van Leeuwen. J. P., Wagter. H. and Oxman. R. M. (1995), A Feature Based Approach to Modelling Architectural Information, *CIB Workshop on Computers and Information in Construction, Modelling of Buildings Through Their Life Cycle*, M. Fischer, K. H. Law and B. Luiten (eds.), August 21-23 1995, Stanford, California, pp. 260-269.

Van Nederveen. S. (1995), *COMBINE IDM*, Version 3.0.1, Source on Internet at http://dutcu15.tudelft/../core_2.htm.

Vanier. D. and Grabinsky. M. W. (1987), Integrated Computer-Aided Building Design Systems, *ARECDA087 Conference Proceedings*, ITEC, Barcelona, pp.34-37.

Vanier. D. J., Mellon. B. S., Thomas. R. and Worling. J. L. (1993), Management of Construction Information Technology, *Proceedings of the First International Conference of the Management of Information Technology for Construction*, Mathur. K. S., Betts. M. P. and Tham. K. W. (eds.), National University of Singapore, August, pp. 75-84.

Vienna University of Technology (1997), *Coordination: S 7000*, on http://www.prip.tuwien.ac.at/lsp/coord/coord.html.

Vincent. S. (1995), Integrating Different Views of Integration, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, pp. 53 - 69.

Wager. D. M. and Scoins. D. (1984), *More Computer Programs for Costruction Industry*, Construction Industry Computing Association (CICA), Cambridge, January.

Ward. M. A. (1996), *ISO/WD 10303-230:1996(E)*, on http://www.leeds.ac.uk/civil/reserach/cae/step/ap_doc/ap_intro.htm

Ward. P.A (1979), *Organisation and Procedures in the Construction Industry*, Macdonald & Evans Ltd., ISBN 0 7121 1530 7.

Watson. A. (1995), Product Models and Beyond, *Integrated Construction Information*, Brandon. P. and Betts. M. (eds.), E & FN Spon, pp. 159 - 172.

Watson. A. and Boyle. A. (1993), Product Models and Application Protocols, *CIVIL-COMP 93, Information Technology for Civil and Structural Engineering*, B. H. V. Topping (ed.), August, Edinburgh, Civil-Comp Press, pp. 121-129.

Wessex (1991), *Wessex SMM7 Building Price Book*, Wessex Electronic Publishing Ltd., Poole, UK.

Westinghouse (1997), *Blackboard Architecture*, Westinghouse Science & Technology Center at http://www.stc.westinghouse.com/tech/area/blackboard. htm.

Winner. R. I. *et al* (1988), *The Role of Concurrent Engineering in Weapons System Acquisition*, IDA Report R-338, Institute of Defense Analysis, Alexandria, Virginia, USA..

Wix. J. and Storer. G. (1995), Product Modelling, *Information Technology Awareness in Engineering, Object Technology and Its Application in Engineering*, J. A. Powell (ed.), March 1995, Salford, UK, pp. 76-85.

Won. D. J. and Klein. M. J. (1989), System Integration through a Database Management System, *ASCE, Proceedings of 6th Conference in Computing in Civil Engineering*, Barnwell. T. Jr. (ed.), September, pp.516-523.

Wright. A. J., Lockley. S. R. and Wiltshire. T. J. (1992), Sharing Data Between Application Programs in Building Design: Product Models and Object-Oriented Programming, *Journal of Building and Environment*, Vol. 27, No. 2, pp. 163-171.

Yamazaki. Y. (1992), Integrated Design and Construction Planning System for Computer Integrated Construction, *Automation in Construction*, Vol.1, No.1, pp. 21-26.

Yang. Y. (1991), *Three-Schema Implementation in STEP*, Product Data Integration Technologies Inc., Technical Report, Los Angeles, California, May 1991.

Zabilski. R. J. and Hall. H. E. (1989), *Presented in 3-D*, Civil Engineering, Vol.59, No.6, pp. 48-50.

Zeid. I. (1991), *CAD/CAM Theory and Practice*, McGraw-Hill, London, ISBN 0-07-072857-7.